

03-31-00

A

03/29/00  
JC777 U.S. PTO

JC625 U.S. PTO  
09/538624  
03/29/00

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Inventorship..... Omoigui et al.  
Applicant..... Microsoft Corporation  
Attorney's Docket No. ....MS1-272USC1  
Title: Managing Timeline Modification and Synchronization of Multiple Media Streams in Networked Client/Server Systems

TRANSMITTAL LETTER AND CERTIFICATE OF MAILING

To: Commissioner of Patents and Trademarks  
Washington, D.C. 20231  
From: Allan T. Sponseller (509) 324-9256  
Lee & Hayes, PLLC  
421 W. Riverside Avenue, Suite 500  
Spokane, WA 99201

The following enumerated items accompany this transmittal letter and are being submitted for the matter identified in the above caption.

- 1. Transmittal Letter with Certificate of Mailing included.
- 2. PTO Return Postcard Receipt
- 3. Check in the Amount of \$1,546.00
- 4. Fee Transmittal
- 5. New patent application (title page plus 51 pages, including claims 1-48 & Abstract)
- 6. Executed Declaration
- 7. 12 sheets of formal drawings (Figs. 1-14)
- 8. Assignment w/Recordation Cover Sheet

Large Entity Status [x]      Small Entity Status [ ]

The Commissioner is hereby authorized to charge payment of fees or credit overpayments to Deposit Account No. 12-0769 in connection with any patent application filing fees under 37 CFR 1.16, and any processing fees under 37 CFR 1.17.

Date: 3/29/00  
By: [Signature]  
Allan T. Sponseller  
Reg. No. 38,318

CERTIFICATE OF MAILING

I hereby certify that the items listed above as enclosed are being deposited with the U.S. Postal Service as either first class mail, or Express Mail if the blank for Express Mail No. is completed below, in an envelope addressed to The Commissioner of Patents and Trademarks, Washington, D.C. 20231, on the below-indicated date. Any Express Mail No. has also been marked on the listed items.

Express Mail No. (if applicable) EL580803625

Date: 3/29/2000  
By: [Signature]  
Lori A. Vierra

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION FOR LETTERS PATENT

**Managing Timeline Modification and Synchronization  
of Multiple Media Streams in Networked Client/Server  
Systems**

Inventor(s):  
Nosakhare D. Omoigui  
Anoop Gupta

1 **RELATED APPLICATIONS**

2 This is a continuation-in-part of U.S. Patent Application No. 09/153,664,  
3 filed September 15, 1998, entitled "Multimedia Timeline Modification in  
4 Networked Client/Server Systems".

5  
6 **TECHNICAL FIELD**

7 This invention relates to networked client/server systems and to managing  
8 the streaming and rendering of multimedia content in such systems.

9  
10 **BACKGROUND OF THE INVENTION**

11 Multimedia streaming—the continuous delivery of synchronized media  
12 data like video, audio, text, and animation—is a critical link in the digital  
13 multimedia revolution. Today, streaming media is primarily about video and  
14 audio, but a richer, broader digital media era is emerging with a profound and  
15 growing impact on the Internet and digital broadcasting.

16 Synchronized media means multiple media objects that share a common  
17 timeline. Video and audio are examples of synchronized media—each is a  
18 separate data stream with its own data structure, but the two data streams are  
19 played back in synchronization with each other. Virtually any media type can  
20 have a timeline. For example, an image object can change like an animated .gif  
21 file, text can change and move, and animation and digital effects happen over  
22 time. This concept of synchronizing multiple media types is gaining greater  
23 meaning and currency with the emergence of more sophisticated media  
24 composition frameworks implied by MPEG-4, Dynamic HTML, and other media  
25 playback environments.

1 The term "streaming" is used to indicate that the data representing the  
2 various media types is provided over a network to a client computer on a real-  
3 time, as-needed basis, rather than being pre-delivered in its entirety before  
4 playback. Thus, the client computer renders streaming data as it is received from a  
5 network server, rather than waiting for an entire "file" to be delivered.

6 The widespread availability of streaming multimedia enables a variety of  
7 informational content that was not previously available over the Internet or other  
8 computer networks. Live content is one significant example of such content.  
9 Using streaming multimedia, audio, video, or audio/visual coverage of noteworthy  
10 events can be broadcast over the Internet as the events unfold. Similarly,  
11 television and radio stations can transmit their live content over the Internet.

12 Although streaming multimedia content compares favorably with more  
13 traditional paper-based content in most regards, one disadvantage is that it requires  
14 significant time for viewing. It cannot be "skimmed" like paper-based content.  
15 Thus, information consumers are forced to choose between the efficiency of the  
16 written word and the richness of the multimedia experience.

17 The invention described below addresses this disadvantage of prior art  
18 streaming multimedia content, allowing more efficient multimedia perusal of  
19 streaming multimedia presentations than has previously been possible.

## 20 21 **SUMMARY OF THE INVENTION**

22 In a client/server network system, multimedia content is streamed from one  
23 or more servers to the client. The multimedia content includes multiple media  
24 streams that can be streamed to the client from the same server or from different  
25 servers. The user is able to modify the playback speed of the multimedia content,

1 allowing the playback to be either speeded up or slowed down. According to one  
2 aspect of the invention, the multimedia content includes text streams, image,  
3 and/or animation streams.

4 According to one aspect of the invention, a separate control component is  
5 included in the client and/or the server for each individual media stream that  
6 manages the presentation of that particular media stream. An additional master  
7 control component manages the overall timeline modification for all of the streams  
8 in the multimedia content. When a user requests a new playback speed the  
9 timeline of the master control component is changed (either speeded up or slowed  
10 down) in accordance with the user's request. Each of the separate control  
11 components is made aware of this change in the master control timeline, either by  
12 the master control sending messages to the separate controls indicating the change,  
13 or by the separate controls monitoring the master control timeline. Once aware of  
14 the change in the master control timeline, each of the separate control components  
15 can adjust their timelines accordingly.

16 According to another aspect of the invention, the master control detects  
17 when the client/server system will potentially be overloaded (e.g., due to a user  
18 request for a change in playback speed). This overloading can be due to requiring  
19 too much computational power on the part of the client, or on requiring too much  
20 bandwidth between the server and the client. If such an overloading condition  
21 exists, the master control takes appropriate action to avoid the overloading. Such  
22 actions include, for example, changing timescale modification for selected streams  
23 from being performed at the client to being performed at the server, reducing the  
24 quality of selected streams, pausing selected streams, etc.

## **BRIEF DESCRIPTION OF THE DRAWINGS**

Fig. 1 is a block diagram of a networked client/server system in accordance with the invention.

Fig. 2 is a block diagram of a networked computer that can be used to implement either a server or a client in accordance with the invention.

Fig. 3 is a block diagram illustrating communications and rendering of a composite media stream in accordance with the invention.

Fig. 4 is a flowchart illustrating an exemplary process for client-based multimedia time-scale modification.

Fig. 5 is a block diagram illustrating one embodiment in which a plurality of timeline-altered media streams are stored at and provided from a server.

Fig. 6 is a block diagram illustrating another embodiment in which a plurality of timeline-altered media streams are stored at and provided from a server.

Fig. 7 is a block diagram illustrating yet another embodiment in which a plurality of timeline-altered media streams are stored at and provided from a server.

Fig. 8 is a block diagram illustrating yet another embodiment in which a plurality of timeline-altered media streams are stored at and provided from a server.

Fig. 9 is a flowchart illustrating an exemplary process for altering the streaming and time-scale modification of the multimedia content.

Fig. 10 is a block diagram illustrating exemplary sets of timeline correlations between the timelines of media streams.

Fig. 11 is a flowchart illustrating an exemplary process for finding an appropriate presentation time in a new stream when switching from a previous stream to the new stream.

Fig. 12 is a diagrammatic illustration of a graphical user interface window having a time-scale-modification tool for changing a playback speed of streaming multimedia content according to one implementation.

Fig. 13 is a diagrammatic illustration of a graphical user interface window having a time-scale-modification tool according to a second implementation.

Fig. 14 is a diagrammatic illustration of a graphical user interface window having a time-scale-modification tool according to a third implementation.

## **DETAILED DESCRIPTION**

### **General Network Structure**

Fig. 1 shows a client/server network system and environment in accordance with the invention. Generally, the system includes one or more ( $m$ ) network server computers 102, and one or more ( $n$ ) network client computers 104. The computers communicate with each other over a data communications network, which in Fig. 1 includes a public network 106 such as the Internet. The data communications network might also include local-area networks and private wide-area networks. Server computers 102 and client computers 104 communicate with one another via any of a wide variety of known protocols, such as the Hypertext Transfer Protocol (HTTP).

Multimedia servers 102 have access to streaming media content in the form of different media streams. These media streams can be individual media streams (e.g., audio, video, graphical, etc.), or alternatively composite media streams

1 including multiple such individual streams. Some media streams might be stored  
2 as files 108 in a database or other file storage system, while other media streams  
3 110 might be supplied to the server on a “live” basis from other data source  
4 components through dedicated communications channels or through the Internet  
5 itself.

6 The media streams received from servers 102 are rendered at the client  
7 computers 104 as a multimedia presentation, which can include media streams  
8 from one or more of the servers 102. These different media streams can include  
9 one or more of the same or different types of media streams. For example, a  
10 multimedia presentation may include two video streams, one audio stream, and  
11 one stream of graphical images. A user interface (UI) at the client computer 104  
12 allows users to either increase or decrease the speed at which the multimedia  
13 presentation is rendered.

### 14 15 **Streaming Media**

16 In this discussion, streaming media refers to one or more individual media  
17 streams being transferred over a network to a client computer on an as-needed  
18 basis rather than being pre-delivered in their entirety before playback. Each of the  
19 individual media streams corresponds to and represents a different media type and  
20 each of the media streams can be rendered by a network client to produce a user-  
21 perceivable presentation using a particular presentation medium. The individual  
22 media streams can be rendered to produce a plurality of different types of user-  
23 perceivable media, including synchronized audio or sound, video graphics or  
24 motion pictures, animation, textual content, command script sequences, or other  
25 media types that convey time-varying information or content in a way that can be

1 sensed and perceived by a human. The individual media streams have their own  
2 timelines, which are synchronized with each other so that the media streams can  
3 be rendered simultaneously for a coordinated multimedia presentation. These  
4 individual media streams can be delivered to the client computer as individual  
5 streams from one or more servers, as a composite media stream(s) from one or  
6 more servers, or a combination thereof.

7 In this discussion, the term “composite media stream” describes  
8 synchronized streaming data that represents a segment of multimedia content. The  
9 composite media stream has a timeline that establishes the speed at which the  
10 content is rendered. The composite media stream can be rendered to produce a  
11 plurality of different types of user-perceivable media, such as synchronized audio  
12 or sound, video graphics or motion pictures, animation, textual content, command  
13 script sequences, etc. A composite media stream includes a plurality of individual  
14 media streams representing the multimedia content.

15 There are various standards for streaming media content and composite  
16 media streams. The “Advanced Streaming Format” (ASF) is an example of such a  
17 standard, including both accepted versions of the standard and proposed standards  
18 for future adoption. ASF specifies the way in which multimedia content is stored,  
19 streamed, and presented by the tools, servers, and clients of various multimedia  
20 vendors. ASF provides benefits such as local and network playback, extensible  
21 media types, component download, scalable media types, prioritization of streams,  
22 multiple language support, environment independence, rich inter-stream  
23 relationships, and expandability. Further details about ASF are available from  
24 Microsoft Corporation of Redmond, Washington.

1           Regardless of the streaming format used, an individual data stream contains  
2 a sequence of digital data units that are rendered individually, in sequence, to  
3 produce an image, sound, or some other stimuli that is perceived by a human to be  
4 continuously varying. For example, an audio data stream comprises a sequence of  
5 sample values that are converted to a pitch and volume to produce continuously  
6 varying sound. A video data stream comprises a sequence of digitally-specified  
7 graphics frames that are rendered in sequence to produce a moving picture.

8           For a composite media stream, the individual data streams are typically  
9 interleaved in a single sequence of data packets. Various types of data  
10 compression might be used within a particular data format to reduce  
11 communications bandwidth requirements.

12           The sequential data units (such as audio sample values or video frames) of  
13 the individual streams are associated with both delivery times and presentation  
14 times, relative to an arbitrary start time. The delivery time of a data unit indicates  
15 when the data unit should be delivered to a rendering client. The presentation time  
16 indicates when the value should be actually rendered. Normally, the delivery time  
17 of a data unit precedes the presentation time.

18           The presentation times determine the actual speed of playback. For data  
19 streams representing actual events or performances, the presentation times  
20 correspond to the relative times at which the data samples were actually recorded.  
21 The presentation times of the various different individual data streams are  
22 consistent with each other so that the streams remain coordinated and  
23 synchronized during playback.  
24  
25

## Exemplary Computer Environment

In the discussion below, the invention will be described in the general context of computer-executable instructions, such as program modules, being executed by one or more conventional personal computers. Generally, program modules include routines, programs, objects, components, data structures, etc. that perform particular tasks or implement particular abstract data types. Moreover, those skilled in the art will appreciate that the invention may be practiced with other computer system configurations, including hand-held devices, multiprocessor systems, microprocessor-based or programmable consumer electronics, network PCs, minicomputers, mainframe computers, and the like. In a distributed computer environment, program modules may be located in both local and remote memory storage devices.

Alternatively, the invention could be implemented in hardware or a combination of hardware, software, and/or firmware. For example, one or more application specific integrated circuits (ASICs) could be programmed to carry out the invention.

Fig. 2 shows a general example of a computer 142 that can be used in accordance with the invention. Computer 142 is shown as an example of a computer that can perform the functions of any of client computers 102 or server computers 104 of Fig. 1.

Computer 142 includes one or more processors or processing units 144, a system memory 146, and a system bus 148 that couples various system components including the system memory 146 to processors 144.

The bus 148 represents one or more of any of several types of bus structures, including a memory bus or memory controller, a peripheral bus, an

1 accelerated graphics port, and a processor or local bus using any of a variety of  
2 bus architectures. The system memory includes read only memory (ROM) 150  
3 and random access memory (RAM) 152. A basic input/output system (BIOS) 154,  
4 containing the basic routines that help to transfer information between elements  
5 within computer 142, such as during start-up, is stored in ROM 150. Computer  
6 142 further includes a hard disk drive 156 for reading from and writing to a hard  
7 disk, not shown, a magnetic disk drive 158 for reading from and writing to a  
8 removable magnetic disk 160, and an optical disk drive 162 for reading from or  
9 writing to a removable optical disk 164 such as a CD ROM or other optical media.  
10 The hard disk drive 156, magnetic disk drive 158, and optical disk drive 162 are  
11 connected to the system bus 148 by an SCSI interface 166 or some other  
12 appropriate interface. The drives and their associated computer-readable media  
13 provide nonvolatile storage of computer readable instructions, data structures,  
14 program modules and other data for computer 142. Although the exemplary  
15 environment described herein employs a hard disk, a removable magnetic disk 160  
16 and a removable optical disk 164, it should be appreciated by those skilled in the  
17 art that other types of computer readable media which can store data that is  
18 accessible by a computer, such as magnetic cassettes, flash memory cards, digital  
19 video disks, random access memories (RAMs) read only memories (ROM), and  
20 the like, may also be used in the exemplary operating environment.

21 A number of program modules may be stored on the hard disk, magnetic  
22 disk 160, optical disk 164, ROM 150, or RAM 152, including an operating system  
23 170, one or more application programs 172, other program modules 174, and  
24 program data 176. A user may enter commands and information into computer  
25 142 through input devices such as keyboard 178 and pointing device 180. Other

1 input devices (not shown) may include a microphone, joystick, game pad, satellite  
2 dish, scanner, or the like. These and other input devices are connected to the  
3 processing unit 144 through an interface 182 that is coupled to the system bus. A  
4 monitor 184 or other type of display device is also connected to the system bus  
5 148 via an interface, such as a video adapter 186. In addition to the monitor,  
6 personal computers typically include other peripheral output devices (not shown)  
7 such as speakers and printers.

8 Computer 142 operates in a networked environment using logical  
9 connections to one or more remote computers, such as a remote computer 188.  
10 The remote computer 188 may be another personal computer, a server, a router, a  
11 network PC, a peer device or other common network node, and typically includes  
12 many or all of the elements described above relative to computer 142, although  
13 only a memory storage device 190 has been illustrated in Fig. 2. The logical  
14 connections depicted in Fig. 2 include a local area network (LAN) 192 and a wide  
15 area network (WAN) 194. Such networking environments are commonplace in  
16 offices, enterprise-wide computer networks, intranets, and the Internet. In the  
17 described embodiment of the invention, remote computer 188 executes an Internet  
18 Web browser program such as the "Internet Explorer" Web browser manufactured  
19 and distributed by Microsoft Corporation of Redmond, Washington.

20 When used in a LAN networking environment, computer 142 is connected  
21 to the local network 192 through a network interface or adapter 196. When used  
22 in a WAN networking environment, computer 142 typically includes a modem 198  
23 or other means for establishing communications over the wide area network 194,  
24 such as the Internet. The modem 198, which may be internal or external, is  
25 connected to the system bus 148 via a serial port interface 168. In a networked

1 environment, program modules depicted relative to the personal computer 142, or  
2 portions thereof, may be stored in the remote memory storage device. It will be  
3 appreciated that the network connections shown are exemplary and other means of  
4 establishing a communications link between the computers may be used.

5 Generally, the data processors of computer 142 are programmed by means  
6 of instructions stored at different times in the various computer-readable storage  
7 media of the computer. Programs and operating systems are typically distributed,  
8 for example, on floppy disks or CD-ROMs. From there, they are installed or  
9 loaded into the secondary memory of a computer. At execution, they are loaded at  
10 least partially into the computer's primary electronic memory. The invention  
11 described herein includes these and other various types of computer-readable  
12 storage media when such media contain instructions or programs for implementing  
13 the steps described below in conjunction with a microprocessor or other data  
14 processor. The invention also includes the computer itself when programmed  
15 according to the methods and techniques described below. Furthermore, certain  
16 sub-components of the computer may be programmed to perform the functions  
17 and steps described below. The invention includes such sub-components when  
18 they are programmed as described. In addition, the invention described herein  
19 includes data structures, described below, as embodied on various types of  
20 memory media.

21 For purposes of illustration, programs and other executable program  
22 components such as the operating system are illustrated herein as discrete blocks,  
23 although it is recognized that such programs and components reside at various  
24 times in different storage components of the computer, and are executed by the  
25 data processor(s) of the computer.

## **Client-Based Multimedia Time-Scale Modification**

As shown in Fig. 1, a network system in accordance with the invention includes a network server(s) 102 from which a plurality of media streams are available. In some cases, the media streams are actually stored by server(s) 102. In other cases, server(s) 102 obtain the media streams from other network sources or devices.

The system also includes network clients 104. Generally, the network clients 104 are responsive to user input to request media streams corresponding to selected multimedia content. In response to a request for a media stream corresponding to multimedia content, server(s) 102 streams the requested media streams to the network client 104 in accordance with some known format such as ASF. The client renders the data streams to produce the multimedia content.

A network client 104 also accepts a speed designation from a human user. The speed designation is a speed factor relative to the original or default playback speed of the selected multimedia content. For example, a speed factor of 1.2 indicates that the multimedia content is to be rendered at 1.2 times its original or default speed, thereby achieving time compression. A speed factor of 0.8 indicates that the multimedia content is to be rendered at 0.8 times its original or default speed, thereby achieving time expansion.

In response to the speed designation from the user, the system modifies the timelines of the individual media streams of the multimedia content, while keeping the timelines synchronized with each other and while maintaining the original pitch of any audio produced from audio streams. In one embodiment of the invention, such timeline modification is performed by the network client. In other

embodiments of the invention, the timeline modification can be performed at the network server before the media streams are streamed to the network client.

Timeline modification changes the timeline of the received data streams in accordance with the user speed designation to achieve either time compression or time expansion (also referred to as “time-scale modification”). With some types of media, such as video, text, and image streams, this involves omitting selected frames or modifying the presentation times of the individual data units or video frames. In other cases, such as with audio streams, the time-modification is more difficult—simply changing the presentation times would alter the pitch of the original audio and make it unintelligible. Accordingly, some type of audio processing technique is used to time-compress or time-expand audio streams, while maintaining the original pitch of the audio—thereby maintaining the intelligibility of the audio.

There are various known methods of audio time modification, commonly referred to as “time-scale-modification,” most of which concentrate on removing redundant information from the speech signal. In a method referred to as sampling, short segments are dropped from the speech signal at regular intervals. Cross fading or smoothing between adjacent segments improves the resulting sound quality.

Another method, referred to as synchronized overlap add method (SOLA or OLA), consists of shifting the beginning of a new speech segment over the end of the preceding segment to find the point of highest cross-correlation (i.e., maximum similarity). The overlapping frames are averaged, or smoothed together, as in the sampling method.

1 Sampling with dichotic presentation is a variant of the sampling method  
2 that takes advantage of the auditory system's ability to integrate information from  
3 both ears. In improves on the sampling method by playing the standard sampled  
4 signal to one ear and the "discarded" material to the other ear. Intelligibility and  
5 compression increase under this dichotic presentation condition when compared  
6 with standard presentation techniques.

7 The methods mentioned above are considered "linear" because all portions  
8 of the speech signal are compressed or expanded uniformly. Other methods are  
9 considered non-linear because they non-uniformly remove portions of the time  
10 signal. One example of a non-linear time-compression method is referred to as  
11 pause removal. When using this method, a speed processing algorithm attempts to  
12 identify and remove any pauses in a recording. Either linear or non-linear time-  
13 scale modification can be used with the invention.

14 More information regarding audio time modification is given in an article  
15 that appeared in the March, 1997, issue of "ACM Transactions on Computer-  
16 Human Interaction" (Volume 4, Number 1, pages 3-38) (1997). For purposes of  
17 this disclosure, it can be assumed that audio time modification involves some  
18 combination of changing individual data stream samples, dropping certain  
19 samples, and adjusting presentation times of any samples that are actually  
20 rendered.

21 Similarly, text streams can also be time-scale modified either linearly or  
22 non-linearly. Linear time-scale modification can be accomplished by speeding up  
23 or slowing down the rate at which the text data is streamed to the client and/or  
24 rendered by the client. Non-linear time-scale modification can be accomplished  
25 by using an algorithm to summarize the text data by selecting key words, phrases,

1 sentences or paragraphs. There are various known methods for selecting such  
2 words or portions of textual content, such as the term frequency/inverse document  
3 frequency technique.

4 Time-scale modification of image streams can also be performed linearly or  
5 non-linearly. Linear time-scale modification can be accomplished by speeding up  
6 or slowing down the rate at which the image data is streamed to the client and/or  
7 rendered by the client. Non-linear time-scale modification can be accomplished  
8 by using an algorithm to analyze the images and rank their importance relative to  
9 one another. Less important images can then be removed to time-compress the  
10 image stream. There are various known methods of determining the importance of  
11 images, such as the compressed domain shot boundary detection, pixel-based shot  
12 boundary detection, histogram-based shot boundary detection, and feature-based  
13 shot boundary detection algorithms.

14 Non-linear time-scale modification of image streams can also be  
15 accomplished by using progressive rendering. In progressive rendering, each  
16 image is made up of multiple layers that are streamed to the client. These layers  
17 are rendered at the client with subsequent layers being overlaid on top of previous  
18 layers, each subsequent layer providing further detail to the previous layers. The  
19 image stream can thus be time modified by removing (or adding) layers to the  
20 images.

21 Animation streams are similar to image streams, except that the images of  
22 an animation stream are tied to a timeline. Animation streams can be linearly or  
23 non-linearly time-scale modified. Linear time-scale modification can be  
24 accomplished by speeding up or slowing down the timeline the images in the  
25 animation stream are tied to, thereby reducing or increasing the duration that each

1 image in the animation stream is rendered. Non-linear time-scale modification can  
2 also be accomplished using any of the techniques discussed above with reference  
3 to image streams.

4 Fig. 3 illustrates an embodiment of the invention in which timeline  
5 modification is performed by network client 104. Network server 102 streams a  
6 composite media stream 202 to network client 104 (although not shown in Fig. 3,  
7 other communications also take place bi-directionally between server 102 and  
8 client 104, such as control-oriented communications). The composite media  
9 stream 202 has a plurality of individual media streams as described above. For  
10 purposes of discussion, it is assumed in this example that the composite media  
11 stream has an audio stream, two video streams, a text stream, an image stream, and  
12 an animation stream. In the illustrated example, the media streams are received  
13 from a single server as a composite media stream. Alternatively, the media  
14 streams may be received as individual (or composite) streams from multiple  
15 servers.

16 Each media stream has a timeline, and the timelines of the individual  
17 streams are synchronized with each other so that the streams can be rendered in  
18 combination to produce coordinated multimedia content at the network client 104.  
19 The original timelines correspond to the original recording or rendition of the  
20 multimedia material, so that rendering the streams according to their timelines  
21 results in presentation speeds that closely match the speed of the original event or  
22 performance. In the case of audio streams, the timelines preserve the original  
23 speed and pitch of the original audio content.

24 The client computer has a demultiplexer component 204 that receives the  
25 composite media stream and that separates out the individual media streams from

1 the composite format in which the data is streamed (such as ASF). This results in  
2 video streams 206 and 208, an audio stream 210, a text stream 212, an image  
3 stream 214, and an animation stream 216. Client 104 includes a different  
4 “control” 218, 220, 222, 224, 226, and 228 for each of the media streams 206, 208,  
5 210, 212, 214, and 216, respectively. Each of these controls is a set of  
6 instructions, executed by a processor of client 104, that manages the presentation  
7 of its corresponding media stream. Client 104 also includes a master control 230  
8 that coordinates the overall presentation of the media content, as discussed in more  
9 detail below.

10 The individual media streams are sent to and received by respective  
11 decoders 232, 234, 236, 238, 240, and 242 that perform in accordance with the  
12 particular data format being employed. For example, the decoders might perform  
13 data decompression.

14 The decoded data streams are then sent to and received by respective time  
15 modification components: video timeline modification components 244 and 246,  
16 an audio timeline modification component 248, a text timeline modification  
17 component 248, an image timeline modification component 252, and an animation  
18 timeline modification component 254. These components receive input from a  
19 human operator in the form of a speed designation as described above. The  
20 timeline modification components change the timelines of the received media  
21 streams in accordance with the user speed designation to achieve either linear time  
22 compression or linear time expansion. With some types of media (e.g., with video  
23 streams, text streams, image streams, or animation streams) this involves either  
24 omitting selected portions of the streams or modifying the presentation times of  
25 the individual data units or frames of the stream. In other cases (e.g., with audio

streams), some type of audio processing technique as the SOLA technique described above is used to time-compress or time-expand audio streams, while maintaining the original pitch of the audio and to also retain the intelligibility of the audio.

The timeline modification components 244 – 254 produce individual media streams that are provided to and received by respective renderers 256, 258, 260, 262, 264, and 266. The rendering components 256 – 266 render the streams in accordance with their modified timelines, as the streams continue to be streamed from the network server. In alternative embodiments of the invention, timeline modification components 244 – 254 might be eliminated and their functions performed by decoders 232 – 242.

Note that the speed designation, provided by the user, dictates the rate at which the network client consumes the composite data stream. Because of this, the client communicates the speed designation to the network server when requesting a particular media stream. The server responds by streaming the media stream at a rate that depends on or is proportional to the speed designation provided by the user. For example, for a speed factor of 2.0, the client consumes data at twice the normal rate. Accordingly, the server streams the media stream at twice its normal rate to meet the demands of the client.

In the described embodiment, the user is allowed to change the speed designation during rendering of the multimedia content. In some cases, however, it may not be possible to change the playback speed without interrupting the playback momentarily. If this is the case, playback resumes as soon as possible, beginning at a point that shortly precedes the point at which playback was discontinued. Thus, there is some overlap in the presentation—when the

1 presentation resumes, the overlap provides context for the new content that  
2 follows.

3 Fig. 4 illustrates an exemplary process for client-based multimedia time-  
4 scale modification. Steps performed at network client 104 are shown on the left-  
5 hand side of the figure, while steps performed by network server 102 are shown on  
6 the right-hand side of the drawing. The process of Fig. 4 may be performed in  
7 software, and is described with additional reference to components in Fig. 3.

8 Multimedia content from network server 102 is selected for rendering at  
9 network client 102 (step 282). In most cases, a user performs this selection from a  
10 menu of available content or via an URL (uniform resource locator) selection. In  
11 some cases, different media streams might be available for a particular content  
12 segment, varying perhaps in quality and in required bandwidth. Preferably,  
13 however, the user is unaware of anything except the simple act of selecting a  
14 single topic or composite stream.

15 A speed designation for the multimedia content is accepted from a human  
16 user (step 284). This step is independent of the previous step of selecting the  
17 content itself. Furthermore, the user can vary the speed designation at any time  
18 during presentation of the selected content, without having to re-select the content.

19 The selected content is requested from the server at a speed that will satisfy  
20 the requirements of the user's speed designation (step 286). Based on this request,  
21 the server identifies the particular composite media stream corresponding to the  
22 selected content (step 288). The server streams this composite media stream to the  
23 client (step 290). In this embodiment, the composite media stream has its original  
24 timeline, which does not necessarily result in the speed that the user has  
25 designated for playback.

1 The client receives the streaming content (step 292) and modifies the  
2 timeline of the media stream(s) in accordance with the speed designation provided  
3 by the user (step 294). As described above, this involves modifying the timelines  
4 of the individual media streams while maintaining their synchronization and  
5 intelligibility. The composite media stream is then rendered in accordance with its  
6 modified timeline (step 296).

### 8 **Server-Based Multimedia Time-Scale Modification**

9 In various embodiments of the invention, modifying the timeline of the  
10 requested multimedia content can be performed dynamically (or “on the fly”) in  
11 the client as described above, in the server, or in both the client and server. In  
12 embodiments where the timeline modification for a stream is carried out at the  
13 server the time modification component 244 – 254 of Fig. 3 for that stream need  
14 not be included in client 104. Rather, components 244 – 254 would be included in  
15 the corresponding server that is providing the stream. Additionally, modifying the  
16 timeline of different streams for requested multimedia content can be performed in  
17 different locations for different streams. For example, audio and video timeline  
18 modification may be performed at the server, while image, animation, and text  
19 timeline modification may be performed at the client. However, in the network  
20 environment, it is often desirable to avoid performing any significant timeline  
21 modification in the server. Otherwise, the server could quickly become  
22 overloaded with requests from multiple clients.

23 Alternatively, in some cases it may be desirable to store multiple versions  
24 of media streams at a server and to select particular versions of the media streams  
25 depending on the timeline requirements of the client, as designated by the user.

1 One advantage of this method is that it can require comparatively less  
2 communications bandwidth between the server and client.

3 As a general example, a server might store a plurality of media streams  
4 having timelines modified by different factors. When a client requests a  
5 composite media stream, the server selects the version of the media stream whose  
6 timeline most closely accords with the speed designation set by the user. If the  
7 timeline does not exactly match the speed designation, the client can perform  
8 further timeline modification.

9 Fig. 5 illustrates a more specific example. In this embodiment, a server 302  
10 stores multiple media streams 304 corresponding to specific multimedia content  
11 306. The media streams are of different types, such as audio, video, image, and  
12 text. In Fig. 5, audio streams are designated by the letter "A", video streams are  
13 designated by the letter "V", image streams are designated by the letter "I", and  
14 text streams are designated by the letter "T". Any combination of a single audio  
15 stream, a single video stream, a single image stream, and a single text stream can  
16 be rendered to produce the multimedia content.

17 The various individual data streams have timelines that are modified by  
18 different degrees. The speed factors are indicated in Fig. 5. In this embodiment,  
19 the audio, video, text, and image streams are organized as sets, each set forming a  
20 composite media stream having a timeline that has been modified by a factor of  
21 0.5, 1.0, or 1.5.

22 When a client 308 requests multimedia content from server 302, the client  
23 308 identifies both the content and the speed factor. In response, the server selects  
24 the audio, video, image, and text streams that have timelines most closely  
25 approximating the identified speed factor, and combines those individual media

1 streams to form the composite media stream. The resulting composite media  
2 stream is then sent to the client. When the timeline is accelerated, this saves  
3 bandwidth in comparison to sending an unaltered composite media stream having  
4 a higher streaming rate to meet the accelerated consumption demands of the client.

5 As a further optimization, the server can store composite media streams  
6 having different degrees of timeline modification and different degrees of quality.  
7 Generally, a media stream of a lower quality will consume less communications  
8 bandwidth than a media stream of a higher quality. Before selecting an  
9 appropriate media stream, the server determines the available bandwidth between  
10 the server and the client. It then selects a combination of individual media streams  
11 that provides the best quality while requiring no more than the available  
12 bandwidth.

13 Fig. 6 illustrates a further example utilizing this concept. In this case, a  
14 server 310 has stored a single audio stream 312, a single text stream 314, multiple  
15 video streams 316, and multiple image streams 318, all corresponding to a single  
16 multimedia segment 320. The video streams and image streams differ in quality  
17 and corresponding required bandwidth: low (lo), intermediate (med), and high  
18 (hi). However, the video streams and the image streams all have a common,  
19 unmodified timeline.

20 When a client 322 requests the multimedia content from server 310, the  
21 server determines or notes both the speed factor designated by the user and the  
22 available bandwidth. It then selects the video stream that has best available  
23 quality while also requiring no more bandwidth (at the requested speed factor)  
24 than the difference between the available bandwidth and the bandwidth consumed  
25

1 by the selected audio stream. Again, this allows the system to compensate for  
2 various available bandwidths.

3 Fig. 7 shows another example, in which a server 328 has stored multiple  
4 text streams 330 and multiple image streams 332, all corresponding to a single  
5 multimedia segment 334. The text streams differ in the degree by which their  
6 timelines have been modified. In this example, there are text streams having  
7 timelines modified by factors of 0.5, 1.0, and 1.5. The image streams differ in  
8 quality and corresponding required bandwidth: low (lo), intermediate (med), and  
9 high (hi). However, the image streams all have a common, unmodified timeline.

10 When a client 336 requests the multimedia content from server 328, the  
11 server determines or notes both the speed factor designated by the user and the  
12 available bandwidth. It then selects a text stream that most closely accords with  
13 the specified speed factor. It then selects the image stream that has best available  
14 quality while also requiring no more bandwidth than the difference between the  
15 available bandwidth and the bandwidth consumed by the selected text stream.  
16 Again, this allows the system to compensate for various available bandwidths.

17 Fig. 8 illustrates yet another embodiment in which multiple media streams  
18 are stored at the server for use depending upon available bandwidth and upon the  
19 speed factor designated by the user. In this embodiment, a server 340 stores a  
20 single text stream 342 and multiple image streams 344, all corresponding to and  
21 representing the same multimedia content 346. The text stream has an unaltered  
22 timeline. However, the image streams have different timelines and also vary by  
23 quality and corresponding bandwidth requirements. Specifically, in this example  
24 the image streams have timelines modified by factors of 0.5, 1.0, and 2.0. For  
25

1 each speed factor, there is a "low" bandwidth image stream having a relatively low  
2 quality, and a "high" bandwidth image stream having a relatively high quality.

3 At a normal, unaltered playback rate, assume the text stream utilizes a  
4 bandwidth of 16 Kbps (kilobits per second), the low bandwidth image streams  
5 require a bandwidth of 20 Kbps, and while the high bandwidth image streams  
6 require a bandwidth of 40 Kbps. Now, suppose that a client requests the  
7 multimedia content over a communications channel having a bandwidth of 56  
8 Kbps, at a speed factor of 2.0. At this speed factor, the client consumes text data  
9 at twice the normal rate, which in this case is 32 Kbps. That leaves 24 Kbps of  
10 available bandwidth. Accordingly, the server selects the low bandwidth image  
11 stream with the timeline modified by a factor of 2.0, and combines it with the text  
12 stream to form a composite media stream for streaming to the client. The total  
13 required communications bandwidth is 52 Kbps, which is within the limits of the  
14 available bandwidth.

15 Although the example given with reference to Fig. 8 is relatively specific,  
16 this method of bandwidth utilization can be generalized to include other types of  
17 media streams with each stream being assigned a priority.

18 Furthermore, a stream can sometimes be timeline-modified dynamically at  
19 the server without incurring significant overhead. Accordingly, the server can  
20 adjust the timeline and quality of the stream dynamically to match the available  
21 bandwidth, eliminating the need to store multiple streams of the same content at  
22 the server. As an example of a situation where this might be easily accomplished,  
23 an MPEG (Motion Picture Expert Group) video stream contains independent  
24 frames and several levels of dependent frames. One easy way to reduce  
25 bandwidth is to simply drop lower-level dependent frames from the video stream.

1        Additionally, although Figs. 5 – 8 illustrate the streaming of all media  
2 streams for particular multimedia content from a single server, the streams can  
3 alternatively be streamed from multiple servers. Thus, rather than streaming a  
4 composite stream from a single server that includes all the data streams for a  
5 particular multimedia presentation, the individual streams can be received from  
6 different servers. Additionally, multiple composite streams can be received from  
7 different servers, such as a composite stream from one server including image and  
8 text data streams, and another composite stream from another server including  
9 audio and video data streams. Additional communication between servers or  
10 between the client and servers may be required when multiple servers are  
11 streaming the data for particular multimedia content. For example, the text stream  
12 and corresponding required bandwidth selected by one server (or the client) can be  
13 communicated to the other server(s) to allow the other server(s) to determine the  
14 amount of available bandwidth.

### 15 16 **Stream Synchronization**

17        Media content can be provided to the client 104 of Fig. 3 from one or more  
18 servers 102 as discussed above. Different time-scale modification techniques can  
19 be used for each of the different streams. For example, video(1) time modification  
20 component 244 of Fig. 3 may use a non-linear compression algorithm, while  
21 video(2) time modification component 246 may use a linear compression  
22 algorithm. Additionally, time-scale modification for some streams may be carried  
23 out at client 104 and time-scale modification for other streams may be carried out  
24 at the server.

1 Master control 230 of Fig. 3 coordinates the time-scale modification of all  
2 the streams in the multimedia content. Master control 230 receives user requests  
3 for changes in the playback speed of the multimedia content. Such changes are  
4 communicated to the individual stream controls 218 – 228 or to the server(s) that  
5 are providing the time-scale modification (whether it be dynamically modified or  
6 pre-stored streams) for the individual stream(s). Alternatively, the coordination  
7 provided by master control 230 can be distributed partly or wholly throughout  
8 controls 218 – 228, thereby embedding the coordination of presenting a stream  
9 wholly or partly in the control of that stream.

10 When master control 230 receives a user request for a new playback speed  
11 for the multimedia content, master control 230 sends a message to each of the  
12 individual stream controls 218 – 228 of the new playback speed. This message is  
13 used by the corresponding time modification components 244 – 254 (whether they  
14 be located in the client 104 or server 102) to change the time-scale modification  
15 being performed to the new playback speed.

16 In an alternate embodiment, master control 230 does not send such  
17 messages to the individual stream controls 218 – 228. Rather, master control 230  
18 maintains a presentation clock referred to as the “master clock”. Each of the  
19 individual stream controls 218 – 228 maintains its own clock, referred to as a  
20 “slave clock”, that the respective controls 218 – 228 synchronize with the master  
21 clock. The controls 218 – 228 monitor the master clock and keep their slave  
22 clocks in pace with the master clock, speeding up or slowing down their respective  
23 slave clocks as the master clock speeds up or slows down.

24 By maintaining a master clock and slave clock relationship, each of the  
25 controls 218 – 228 is alleviated of the burden of providing “elegant” time-scale

1 modification. That is, some of the controls 218 – 228 may not have the ability to  
2 speed up or slow down the rate at which the media stream is rendered, remove less  
3 important portions of the media stream, etc. Rather, these controls may merely be  
4 able to detect when they are out of synchronization with the master clock and  
5 either jump ahead in their rendering or temporarily pause their rendering until they  
6 are re-synchronized.

7 Master control 230 may also perform additional monitoring of the  
8 multimedia content and alter the time-scale modification being performed based  
9 on available bandwidth between the server and client and/or based on the  
10 processing capabilities of the client.

11 Fig. 9 illustrates an exemplary process used by master control 230 to alter  
12 the streaming and time-scale modification for the multimedia content. The  
13 process of Fig. 9 may be performed in software. Fig. 9 is described with  
14 additional reference to components in Fig. 3.

15 Master control 230 monitors the usage of both the bandwidth between  
16 server 102 and client 104 and the processing capabilities of client 104 (step 352).  
17 Master control 230 can be either pre-programmed or dynamically programmed  
18 with the server to client bandwidth devoted to streaming the multimedia content  
19 and the processing capabilities of client 104 devoted to playing back the  
20 multimedia content. Master control 230 compares these programmed values to the  
21 current bandwidth and processing usage to determine whether to make an  
22 alteration in a stream(s). The monitoring of step 352 can be performed  
23 continually, or alternatively in response to certain events (such as a new playback  
24 speed being requested by the user).

1       The monitored and programmed values are used to determine whether the  
2 bandwidth allotted to streaming the data or the processing capacity has been  
3 exceeded (step 354). Such changes can result, for example, due to a user request  
4 for a faster playback speed, or a reduction in the amount of bandwidth or  
5 processing capacity that can be devoted to streaming or playing back of the  
6 multimedia content.

7       If the allotted bandwidth or processing capacity has been exceeded, then  
8 master control 230 selects a stream(s) to be altered (step 356). The selection of  
9 stream(s) can be accomplished in a variety of manners. An ordered list can be  
10 provided to the master control (e.g., generated by the author of master control 230,  
11 by the author of the multimedia content or the user of client 104) that identifies the  
12 order in which streams are to be selected. Alternatively, each stream may be given  
13 a priority ranking and this priority ranking used by master control 230 to  
14 determine which stream to select for alteration (e.g., the lowest priority stream).

15       Master control also alters the selected stream(s) to conform to the current  
16 bandwidth and processing capacity requirements (step 358). In the illustrated  
17 example this alteration includes one or more of transferring time-scale  
18 modification for a selected stream(s) from client 104 to server 102, reducing the  
19 quality of the selected stream(s), or pausing a selected stream(s). Which of these  
20 actions is to be performed by master control can be determined by a set of rules  
21 programmed into master control 230. These rules can be generated by, for  
22 example, the author of master control 230, the author of the multimedia content, or  
23 the user of client 104. For example, the rules may indicate that all streams should  
24 continue to be played back regardless of the quality reduction of the selected  
25 streams, the rules may indicate that time-modification of only certain streams can

1 be transferred to the client, the rules may indicate that audio or text streams should  
2 be paused rather than reducing the quality of any of the other streams, etc. Master  
3 control 230 sends messages to the appropriate individual stream controls as well as  
4 the appropriate servers to change the quality of a stream, pause a stream, or  
5 transfer time-scale modification processing from the local stream control to the  
6 server.

7         Returning to step 352, master control 230 also checks whether there is  
8 excess bandwidth or processing capacity that it can use (step 360). Such excess  
9 bandwidth or processing capacity can arise, for example, due to a reduction in the  
10 playback speed of the multimedia content or extra capacity or bandwidth being  
11 devoted to streaming or playback of the multimedia content. If such excess  
12 bandwidth or processing capacity is detected, master control 230 selects a  
13 stream(s) to alter (step 362). This selection process is analogous to that of step  
14 356, except that the ordering of streams may be “reversed”. For example, higher  
15 priority streams may be selected for improved quality due to the excess bandwidth  
16 or processing capacity. Alternatively, master control 230 may select the same  
17 streams that were previously selected for alteration in step 356.

18         Master control 230 also alters the selected stream(s) to take advantage of  
19 the excess bandwidth or processing capacity. This alteration is analogous to that  
20 of step 358, or alternatively may be to “undo” whatever alteration was previously  
21 performed for the stream in step 358. In making the selection and alteration in  
22 steps 362 and 364, master control 230 compares the bandwidth and processing  
23 capacities of the proposed alteration to the excess bandwidth or processing  
24 capacity to verify that neither the bandwidth nor the processing capacity devoted  
25 to the multimedia content is exceeded. For example, excess processing capacity at

1 client 104 may be available, but the first alteration that master control 230 would  
2 want to make may exceed the bandwidth constraints and therefore cannot be  
3 carried out. Thus, master control 230 tests another alteration. If no alteration can  
4 be made that violates neither the bandwidth nor the processing capacities, then no  
5 alteration is made.

6 The operation of master control 230 in altering the streaming and the time-  
7 scale modification of the multimedia content is further illustrated in the following  
8 example. Assume that the bandwidth devoted to multimedia content is 150 Kbps  
9 and that the multimedia content includes two video streams, an image stream, an  
10 audio stream, and a text stream. Further assume that the time-scale modification  
11 of each of the video streams is performed at the client, that the time-scale  
12 modification of the image, audio, and text streams is performed at the server, and  
13 that at a speedup factor of 1.0, the video streams each require 30 Kbps, the image  
14 and audio streams each require 20 Kbps, and the text stream requires 10 Kbps. At  
15 the speedup factor of 1.0, the streams require only 110 Kbps of the available 150  
16 Kbps of bandwidth. If the playback speed is increased to a speedup factor of 1.5,  
17 the video streams would require 45 Kbps of bandwidth to be time-compressed at  
18 the client, while the image and audio streams would still require 20 Kbps and the  
19 text stream 10 Kbps as these streams are being time-compressed at the server. At  
20 the speedup factor of 1.5, the streams require only 140 Kbps of the available 150  
21 Kbps of bandwidth. However, if the speedup factor were to be increased to 2.0,  
22 then the video streams would require 60 Kbps while the image, audio, and text  
23 streams would require 20 Kbps, 20 Kbps, and 10 Kbps, respectively. The streams  
24 would require a total of 170 Kbps, which is not available. Thus, the master control  
25 would select and alter at least one of the streams, such as selecting one of the

1 video streams for time-scale modification at the server (which would reduce the  
2 bandwidth requirements to 140 Kbps), or pause the text stream so it is no longer  
3 being streamed (which would reduce the bandwidth requirements to 150 Kbps).

#### 4 5 **Timeline Correlation**

6 When the playback speed of the multimedia content is altered by the user,  
7 the playback of the multimedia content should continue in a relatively  
8 uninterrupted manner, albeit at the new playback speed. For example, suppose a  
9 user changes the playback speed from a speedup factor of 1.0 to a speedup factor  
10 of 1.5. The playback of the multimedia content should continue at approximately  
11 the location where the playback speed change was requested by the user rather  
12 than beginning playback at the beginning of the multimedia content at the new  
13 playback speed.

14 If timeline modification is performed at the client or is performed  
15 dynamically at the server, then the client or server can begin timeline modification  
16 when the user request is received. As the timeline modification is being  
17 performed “on the fly” on the same underlying data stream, difficulties in  
18 maintaining timeline correlation typically do not arise.

19 However, in situations where multiple versions of a media stream are stored  
20 at a server (e.g., as discussed above with reference to Figs. 5 – 8), timeline  
21 correlation problems can arise. In order for the server to switch from one data  
22 stream to another (corresponding to the new playback speed), the correct location  
23 in the new stream to begin streaming needs to be determined.

24 In order to make such determinations, one of the versions of a particular  
25 data stream (e.g., one of the video streams 304 of Fig. 5) is referred to as a primary

1 or reference version of the media stream. A primary media stream normally has a  
2 timeline that has not been altered. The remaining versions of the data stream  
3 stored by the server are media streams having timelines that have been altered in  
4 accordance with linear and/or non-linear techniques.

5 There is a known timeline correlation between the data units of the various  
6 media streams. The term "timeline correlation" as used herein refers to a  
7 correlation in content between two streams that differ in the degree and/or manner  
8 in which their timelines have been modified. Thus, a playback point one minute  
9 into an unaltered timeline correlates to a point thirty seconds into a timeline that  
10 has been linearly altered by a factor of 2.0 (accelerated to twice the speed of the  
11 original). More generally, the point in the new timeline equals  
12  $\text{oldtime}(\text{oldfactor}/\text{newfactor})$ , where oldtime is the presentation time in the first  
13 media stream at which the speed change is to occur, oldfactor is the playback  
14 speed or factor of the old media stream, and newfactor is the playback speed or  
15 factor of the new media stream.

16 When non-linear timeline alteration is involved, the correlation between  
17 streams cannot be calculated in this manner. In the illustrated example, the  
18 timeline correlations are compiled and stored as the non-linear compression is  
19 performed. The stored data is then referenced by the system when it becomes  
20 necessary to find content in one stream corresponding to the same content in  
21 another stream.

22 Specifically, the server stores one or more sets of timeline correlations  
23 between the timelines of the primary and timeline-altered media streams. These  
24 sets of correlations are arranged to allow each cross-referencing between the  
25 various streams. For example, one set of correlations contains mappings from

1 presentation times of the primary media stream to timeline-correlated presentation  
2 times of the timeline-altered media streams. Other sets of correlations correspond  
3 to individual ones of the time-altered media streams. Each of these sets contains  
4 mappings from presentation times of the corresponding timeline-altered media  
5 stream to correlated presentation times of the primary media stream.

6 Fig. 10 illustrates this more clearly. Shown in Fig. 10 are a primary media  
7 stream 370, a first timeline-altered media stream 372, and a second timeline-  
8 altered media stream 374. In this example, the timeline-altered media streams  
9 have corresponding timelines that are non-linearly altered relative to the timeline  
10 of the primary media stream.

11 Also shown in Fig. 10 are reference tables or data objects corresponding to  
12 the media streams. Table 376, associated with primary media stream 370, is a  
13 cross-reference containing mappings from presentation times of the primary media  
14 stream to timeline-correlated presentation times of the first and second media  
15 streams. Table 376 is indexed by presentation times of the primary media stream.  
16 Thus, for any given presentation time of the primary media stream, it is possible to  
17 quickly find a corresponding or timeline-correlated presentation time in either of  
18 the two timeline-altered media streams.

19 By itself, table 376 is useful when switching from primary media stream  
20 370 to one of the timeline-altered media streams 372 and 374. To transition, for  
21 instance, from the primary media stream to the first timeline-altered media stream,  
22 the current presentation time of the primary media stream is noted. This  
23 presentation time is used as an index into table 376 to find the correlated  
24 presentation time in the first media stream. The first media stream is then initiated  
25 at the correlated time as found in the table.

1 Further tables or data objects 378 and 380 are associated respectively with  
2 first and second timeline-altered media streams 372 and 374, and are used as back-  
3 references to the primary media stream. Each of these tables is indexed by the  
4 presentation times of its associated media stream, to find timeline-correlated  
5 presentation times in the primary media stream.

6 The tables or data objects can be stored and referenced by server 102.  
7 Alternatively, they can be stored by server 102 and downloaded to client 104 as  
8 needed. As a further alternative, the data objects with the timeline-altered media  
9 streams can be provided with individual data units of the timeline-altered media  
10 streams. In accordance with this further alternative, each data unit is accompanied  
11 by a presentation time at which the data unit is to be rendered, and also by a  
12 reference presentation time, where the reference presentation time indicates a  
13 presentation time in the primary reference stream that corresponds to the  
14 presentation time of the data unit in the timeline-altered media stream. This  
15 reference presentation time is then used to index table 376 associated with primary  
16 stream 360.

17 Fig. 11 illustrates the process used to find an appropriate presentation time  
18 in the second timeline-altered media stream, when switching from the first  
19 timeline-altered media stream to the second timeline-altered media stream. The  
20 process of Fig. 11 may be performed in software. Fig. 11 is described with  
21 additional reference to components in Fig. 10.

22 Playback of the first media stream is initially stopped at a particular  
23 presentation time of the first media stream (step 390). A stored table or cross-  
24 reference 378 is referenced to determine a presentation time of the primary media  
25 stream that has a timeline correlation with the particular presentation time at which

1 playback of the first media stream was stopped (step 392). A table 376 of primary  
2 media stream 370 is then referred to in order to determine a presentation time of  
3 the second media stream that has a timeline correlation with the determined  
4 presentation time of the primary media stream (step 394). Playback of the second  
5 media stream is then initiated at a point in the second media stream having a  
6 presentation time that is no greater than the determined presentation time (step  
7 396). In the described embodiment of the invention, playback is initiated  
8 somewhat prior to the determined presentation time, thus providing a short overlap  
9 in the rendered content to provide context when initiating the second timeline-  
10 altered media stream in midstream.

11 The referencing steps are illustrated in Fig. 10. An arrow from the  
12 reference table 378 of first media stream 372 indicates that the table 378 is used to  
13 find a time-correlated presentation time in the primary media stream. This value is  
14 used to index table 376 of primary stream 370 to find a timeline-correlated  
15 presentation time in second media stream 374.

## 17 User Experience

18 The functionality described above is exposed through an application  
19 program executed at network client 104, referred to herein as a streaming  
20 multimedia player. The streaming multimedia player may be incorporated into the  
21 operating system or run as a separate, self-contained application. In either case,  
22 the streaming multimedia player operates in a graphical user interface windowing  
23 environment such as provided by the "Windows" brand of operating systems,  
24 available from Microsoft Corporation of Redmond, Washington..

Fig. 12 shows one implementation of a graphical user interface window 400 for the multimedia player. This UI window 400 has a command bar 402, a media screen 404, shuttle controls 406, a volume control 408, and content information space 410. Command bar 402 lists familiar UI commands, such as “File”, “View”, and so forth.

Media screen 404 is the region of the UI within which the visual media stream(s) is rendered. For video, image, animation, and text streams, the underlying video, images, animations, and text are displayed on screen 404. Each of these streams can be displayed in a different portion of the screen 204 (alternatively, one or more of the portions may be overlapped by another portion).

Shuttle controls 406 enable the user to control play of the multimedia content. Shuttle controls 406 include multiple play buttons 412(1), 412(2), and 412(3), a stop button 414, a pause button 416, rewind buttons 418 and 420, and fast forward buttons 422 and 424.

Play buttons 412(1) – 412(3) are associated with different playback speeds of the multimedia content. In this illustration, play button 412(1) corresponds to a normal playback speed (i.e., “x1.0”), play button 412(2) corresponds to a faster playback speed with a speed up factor of 25% (i.e., “x1.25”), and play button 412(3) corresponds to an even faster playback speed with a speed up factor of 50% (i.e., “x1.50”). It is noted, however, that more or less than three buttons may be used (e.g., two, four, five, etc.) and may correspond to speeds both above and below the normalized speed of “x1.0”.

The user can actuate one of the play buttons via a UI actuation mechanism, such as a pointer 426 or by tabbing to the desired play button and hitting the “enter” key. Upon selection of a play button, the multimedia player plays the

1 multimedia content at the playback speed associated with the selected play button.  
2 For instance, if the user selects play button 412(2) with a 25% speedup factor, the  
3 multimedia player plays the content at a playback speed of 1.25 times the original  
4 or default playback speed.

5 Once the multimedia content is playing at one speed, the user is free to  
6 select a new speed by actuating another of the play buttons 412(1) – 412(3).  
7 Suppose the user decides to slow the content back to normal speed. The user can  
8 actuate the “x1.0” play button 412(1) to return the media content to the normal  
9 speed. In response to speed changes, the multimedia player is configured to repeat  
10 a portion of the multimedia content at the new speed.

11 Content information space 410 lists information pertaining to the  
12 multimedia content being rendered on the media screen 404. The content  
13 information space includes the show name, author and copyright information, and  
14 tracking/timing data.

15 Fig. 13 shows another implementation of a graphical user interface window  
16 440 for the multimedia player. Like UI 400 of Fig. 11, UI 440 has command bar  
17 402, media screen 404, shuttle controls 406, volume control 408, and content  
18 information space 410. This implementation, however, employs only a single play  
19 button 442. Actuation of play button 442 initiates play of the multimedia content.

20 UI 440 has a scale mechanism 444 to vary the speed of the content during  
21 rendering. The scale mechanism has a range of playback speeds 446, which in this  
22 example range from 0.5x to 2.5x the normal speed. Scale mechanism 444 also has  
23 a movable slider 448 that is movable over the range 446. The user can position  
24 the slider 448 at the desired speed at which the multimedia player is to play the  
25 multimedia content.

1 In the Fig. 13 illustration, range 446 is a continuous range from a high  
2 playback speed (i.e., 2.5x) to a low playback speed (i.e., 0.5x). Slider 448 moves  
3 continuously over the range. In other implementations, range 446 is a discrete  
4 range of discrete playback speeds (e.g., 0.5x, 1.0x, 1.5x, 2.0x, and 2.5x) and the  
5 slider is movable among the discrete playback speeds.

6 Once the multimedia content is playing at one speed, the user is free to  
7 select a new speed by moving the slider 448 to a new speed. In response to use  
8 manipulation of the scale mechanism, the multimedia player repeats a portion of  
9 the multimedia content and begins playing at the new speed.

10 Fig. 14 shows a third implementation of a graphical user interface window  
11 460 for the multimedia player. In this implementation, UI 460 has a single play  
12 button 462 to initiate playback of the multimedia content. UI 460 also has a menu  
13 464 associated with the play button. In this illustration, menu 464 is a drop-down  
14 or pull-down menu that opens beneath the play button in response to actuation of a  
15 tab 466 adjacent to the play button. Alternatively, menu 464 may be invoked by  
16 placing pointer 426 over play button 462 and right clicking a mouse button.

17 Menu 464 lists multiple playback speeds from which a user can select. In  
18 the illustrated example, five playback speeds are listed: x0.5, x0.75, x1.0, x1.25,  
19 and x1.5. The user can select one of the listed speeds to instruct the multimedia  
20 player to play the multimedia content at a desired speed. As noted above, the user  
21 can select a new speed after the content has begun playing by invoking the menu  
22 and selecting the new speed. In response, the multimedia player repeats a portion  
23 of the multimedia content and begins playing at the new speed.

24 Although the invention has been described in language specific to structural  
25 features and/or methodological steps, it is to be understood that the invention

1 defined in the appended claims is not necessarily limited to the specific features or  
2 steps described. Rather, the specific features and steps are disclosed as preferred  
3 forms of implementing the claimed invention.

4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25

## CLAIMS

1. A method comprising:

detecting, in a system for streaming a plurality of data streams from a server to a client, a potential overburdening of the system;

selecting at least one of the plurality of data streams in response to detecting the potential overburdening of the system; and

altering playback of the at least one data stream to avoid overburdening the system.

2. A method as recited in claim 1, wherein the detecting comprises detecting a potential overburdening of the system by exceeding a server to client bandwidth devoted to the plurality of data streams.

3. A method as recited in claim 1, wherein the detecting comprises detecting a potential overburdening of the system by exceeding a processing capacity of the client.

4. A method as recited in claim 1, wherein the altering comprises pausing the at least one data stream.

5. A method as recited in claim 1, wherein the altering comprises ceasing time-scale modification of the at least one stream at the client and beginning time-scale modification of the at least one stream at the server.

1           **6.**    A method as recited in claim 1, wherein the altering comprises  
2 reducing a quality of the at least one stream.

3  
4           **7.**    A method as recited in claim 1, wherein the detecting comprises  
5 monitoring the system for the potential overburdening in response to receiving a  
6 new request for a new playback speed for the plurality of data streams.

7  
8           **8.**    A method as recited in claim 1, further comprising:  
9           detecting when excess capacity is available in the system; and  
10           altering playback of at least one of the plurality of data streams in response  
11 to detecting the excess capacity.

12  
13           **9.**    A method as recited in claim 1, further comprising allowing a user to  
14 modify a set of rules used in selecting the at least one of the plurality of data  
15 streams.

16  
17           **10.**   A method as recited in claim 1, further comprising allowing a user  
18 to modify a set of rules used to determine the manner in which playback of the at  
19 least one data stream is altered.

20  
21           **11.**   A method as recited in claim 1, wherein the plurality of data streams  
22 include one or more of an image stream, a text stream, and an animation stream.

1           **12.**     One or more computer-readable memories containing a computer  
2 program that is executable by a processor to perform the method recited in claim  
3 1.

4  
5           **13.**     A system comprising:  
6 a client computer coupled to a network;  
7 a server computer coupled to transmit a plurality of individual data streams  
8 to the client computer via the network; and  
9 wherein the client computer is to detect when bandwidth from the server to  
10 the client computer that is allotted to transmitting the plurality of individual data  
11 streams would be exceeded and take action to prevent the allotted bandwidth from  
12 being exceeded.

13  
14           **14.**     A system as recited in claim 13, wherein the network comprises the  
15 Internet.

16  
17           **15.**     A system as recited in claim 13, wherein the server is to transmit the  
18 plurality of individual data streams to the client computer as a composite media  
19 stream.

20  
21           **16.**     A system as recited in claim 13, wherein the client computer is to  
22 prevent the allotted bandwidth from being exceeded by transferring time-scale  
23 modification responsibility from a control component at the client computer to a  
24 control component at the server computer.

25

1       **17.**    A system as recited in claim 13, wherein the client computer is to  
2 prevent the allotted bandwidth from being exceeded by communicating to the  
3 server computer to cease transmitting one of the plurality of individual data  
4 streams.

5  
6       **18.**    A system as recited in claim 13, wherein the client computer is to  
7 prevent the allotted bandwidth from being exceeded by communicating to the  
8 server computer to switch to a lower-resolution version of one of the plurality of  
9 individual data streams.

10  
11       **19.**    A system as recited in claim 13, wherein the plurality of individual  
12 data streams include one or more of an image stream, a text stream, and an  
13 animation stream.

14  
15       **20.**    A server computer comprising:  
16 a bus;  
17 a memory system, coupled to the bus, to store a plurality of instructions;  
18 and  
19 a processor, coupled to the bus, to execute the plurality of instructions to:  
20       receive an indication that time-scale modification for a data stream  
21 that was previously performed at a client computer should now be  
22 performed at the server computer, and  
23       transmit a time-scale modified data stream to the client computer.

1           **21.**     A server computer as recited in claim 20, wherein the processor is  
2 further to select one of a plurality of pre-stored versions of the data stream to  
3 transmit as the time-scale modified data stream.

4  
5           **22.**     A server computer as recited in claim 20, wherein the processor is  
6 further to generate the time-scale modified data stream by dynamically time-scale  
7 modifying an original version of the data stream.

8  
9           **23.**     A server computer as recited in claim 20, wherein the data stream  
10 comprises one or more of an image stream, a text stream, and an animation stream.

11  
12           **24.**     An apparatus comprising:  
13           a master control component to maintain a master timeline for a multimedia  
14 presentation; and  
15           a plurality of individual stream controls corresponding to individual data  
16 streams for the multimedia presentation, wherein each of the plurality of  
17 individual stream controls is to maintain a timeline for the corresponding  
18 individual data stream.

19  
20           **25.**     An apparatus as recited in claim 24, wherein the master control  
21 component is also to receive a user request for a new playback speed and  
22 communicate the new playback speed to the plurality of individual stream  
23 controls.

24  
25

1           **26.**    An apparatus as recited in claim 25, wherein the master control  
2 component is to communicate the new playback speed to the plurality of  
3 individual stream controls by sending a message to each of the plurality of  
4 individual stream controls.

5  
6           **27.**    An apparatus as recited in claim 24, wherein each of the plurality of  
7 individual stream controls is to monitor the master timeline and adjust the timeline  
8 maintained by the stream control to maintain synchronization with the master  
9 timeline.

10  
11          **28.**    An apparatus as recited in claim 24, wherein the individual data  
12 streams include one or more of an image stream, a text stream, and an animation  
13 stream.

14  
15          **29.**    One or more computer-readable media having stored thereon a  
16 computer program that, when executed by one or more processors, causes the one  
17 or more processors to perform functions including:

18           receiving a user request at a client for a new playback speed of multimedia  
19 content being streamed as a plurality of individual streams to the client; and

20           modifying the playback of each stream of the multimedia content in  
21 accordance with the new playback speed.

1           **30.**   One or more computer-readable media as recited in claim 29,  
2 wherein the computer program further causes the one or more processors to  
3 perform functions including sending a message to each of a plurality of individual  
4 stream controls, the message indicating the new playback speed.

5  
6           **31.**   One or more computer-readable media as recited in claim 30,  
7 wherein the function of sending a message comprises a function of sending the  
8 message to an individual stream control located at a server streaming the  
9 individual stream of the multimedia content.

10  
11           **32.**   One or more computer-readable media as recited in claim 29,  
12 wherein the computer program further causes the one or more processors to  
13 perform functions including each of a plurality of individual stream controls  
14 corresponding to the plurality of individual streams monitoring a master clock and  
15 adjusting a local clock to keep synchronized with the maser clock.

16  
17           **33.**   One or more computer-readable media as recited in claim 29,  
18 wherein the computer program further causes the one or more processors to  
19 perform functions including performing, by an independent stream control located  
20 at the client and corresponding to one of the plurality of individual streams, time-  
21 scale modification of the one stream in accordance with the new playback speed.

1       **34.**    One or more computer-readable media as recited in claim 29,  
2 wherein the multimedia content includes one or more of an image stream, a text  
3 stream, and an animation stream.

4  
5       **35.**    A method comprising:  
6       receiving streaming text from a server;  
7       receiving a user request to change a playback speed of the streaming text;  
8       and  
9       altering the playback speed of the streaming text in accordance with the  
10      user request.

11  
12      **36.**    A method as recited in claim 35, further comprising:  
13      detecting a potential overburdening of a system receiving the streaming  
14      text; and  
15      altering playback of the streaming text to avoid overburdening the system.

16  
17      **37.**    A method as recited in claim 35, wherein the receiving the user  
18      request comprises receiving a user request to increase the playback speed of the  
19      streaming text.

20  
21      **38.**    A method as recited in claim 35, wherein the receiving the user  
22      request comprises receiving a user request to decrease the playback speed of the  
23      streaming text.  
24  
25

1           **39.**    A method as recited in claim 35, wherein the altering comprises  
2 performing linear time-scale modification in accordance with the user request.

3  
4           **40.**    A method as recited in claim 35, wherein the altering comprises  
5 performing non-linear time-scale modification in accordance with the user request.

6  
7           **41.**    One or more computer-readable memories containing a computer  
8 program that is executable by a processor to perform the method recited in claim  
9 35.

10  
11           **42.**    A method comprising:  
12       receiving a plurality of images as streaming image data from a server;  
13       receiving a user request to change a playback speed of the plurality of  
14 images; and  
15       altering the playback speed of the plurality of images in accordance with  
16 the user request.

17  
18           **43.**    A method as recited in claim 42, further comprising:  
19       detecting a potential overburdening of a system receiving the streaming  
20 image data; and  
21       altering playback of the streaming image data to avoid overburdening the  
22 system.

1           **44.**    A method as recited in claim 42, wherein the altering comprises  
2 performing linear time-scale modification in accordance with the user request.

3  
4           **45.**    A method as recited in claim 42, wherein the altering comprises  
5 performing non-linear time-scale modification in accordance with the user request.

6  
7           **46.**    A method as recited in claim 42, further comprising:  
8 receiving each of the plurality of images as a plurality of layers; and  
9 wherein the altering comprises, for each of the plurality of images, reducing  
10 the number of the plurality of layers that are used to render the image.

11  
12           **47.**    A method as recited in claim 42, further comprising receiving  
13 timeline data corresponding to the plurality of images, the timeline data indicating  
14 when the plurality of images are to be rendered.

15  
16           **48.**    One or more computer-readable memories containing a computer  
17 program that is executable by a processor to perform the method recited in claim  
18 42.

1     **ABSTRACT**

2             In a client/server network system, multimedia content is streamed from one  
3 or more servers to the client. The multimedia content includes multiple media  
4 streams that can be streamed to the client from the same server or from different  
5 servers. The user is able to modify the playback speed of the multimedia content,  
6 allowing the playback to be either speeded up or slowed down.

7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25

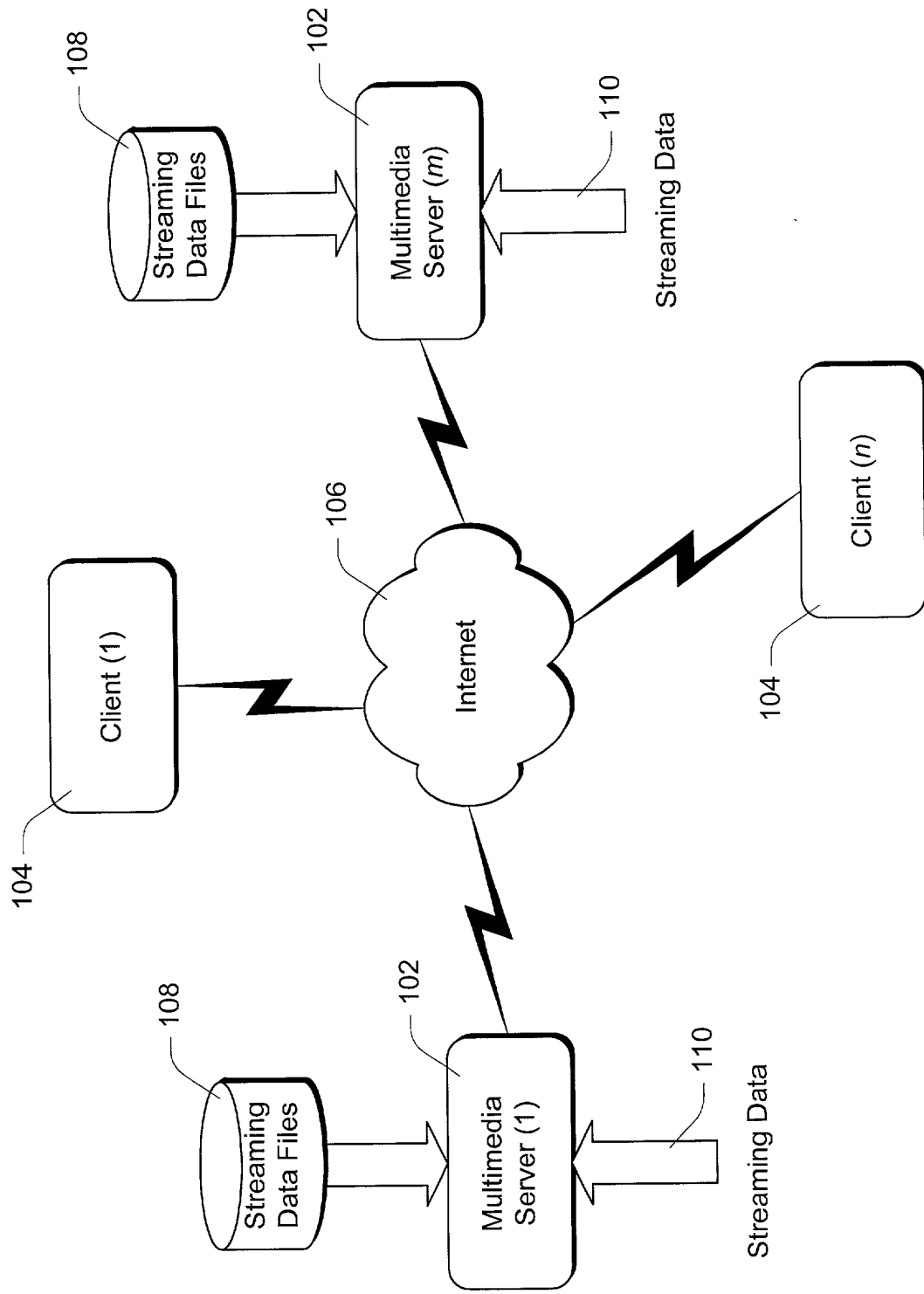
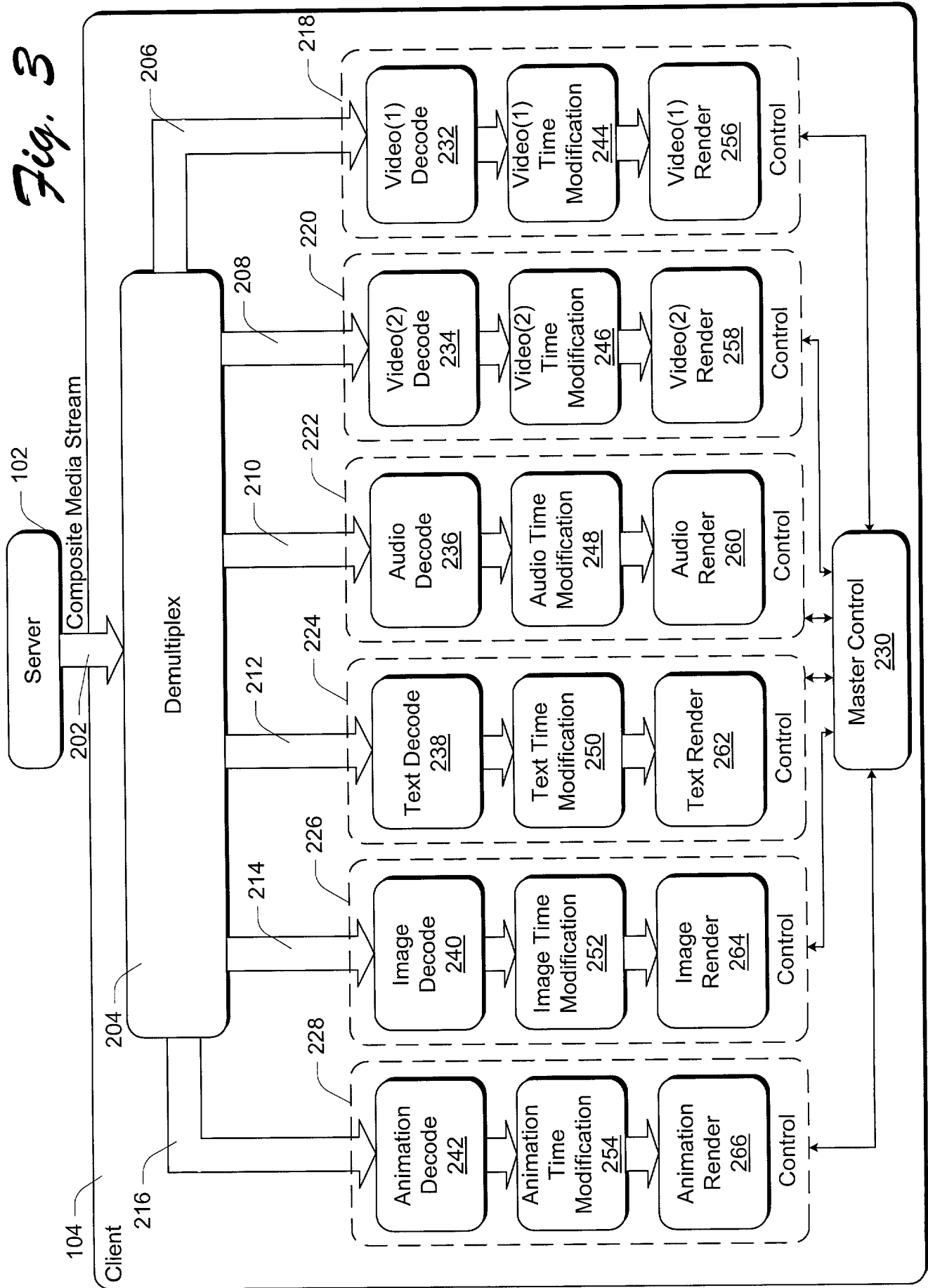
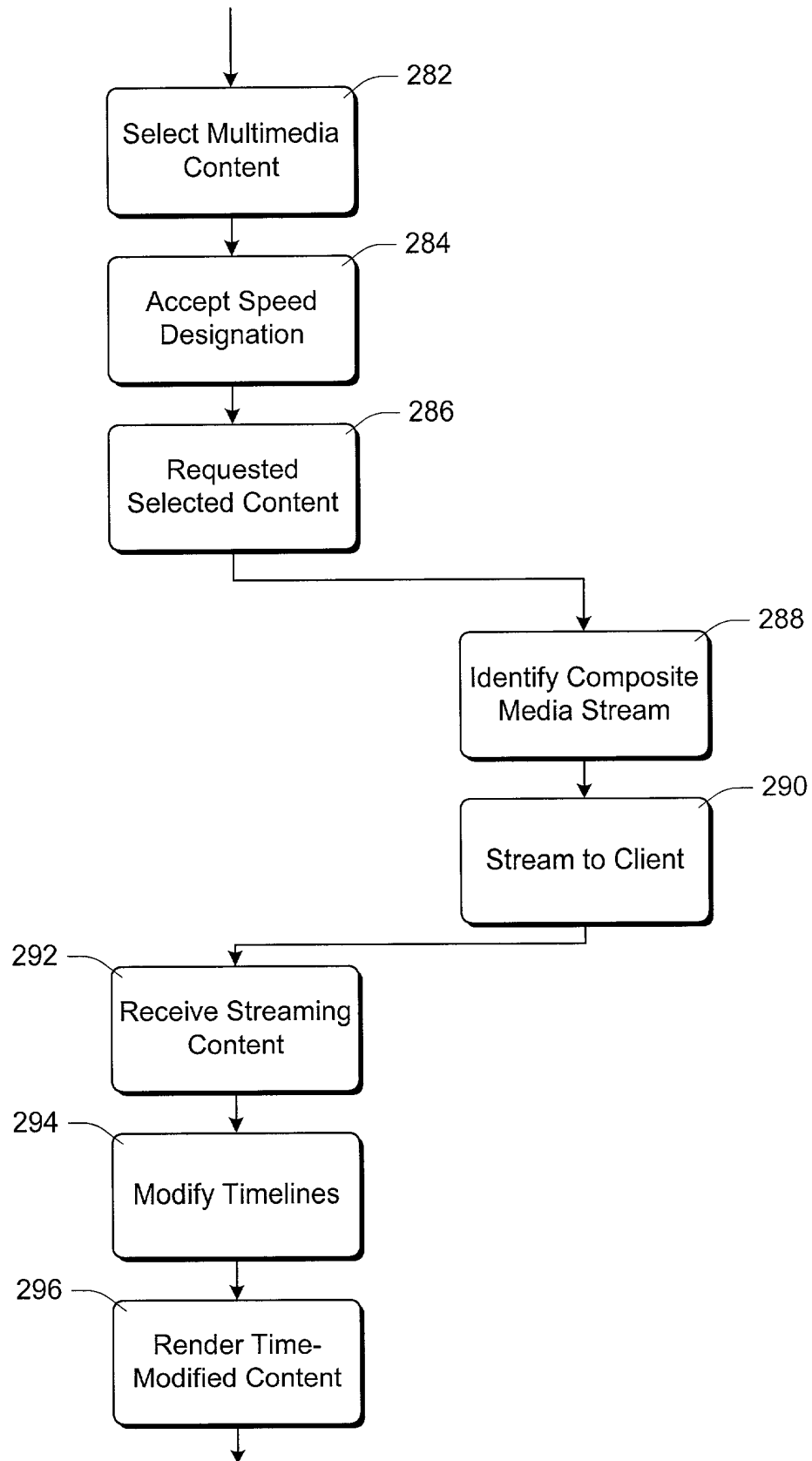
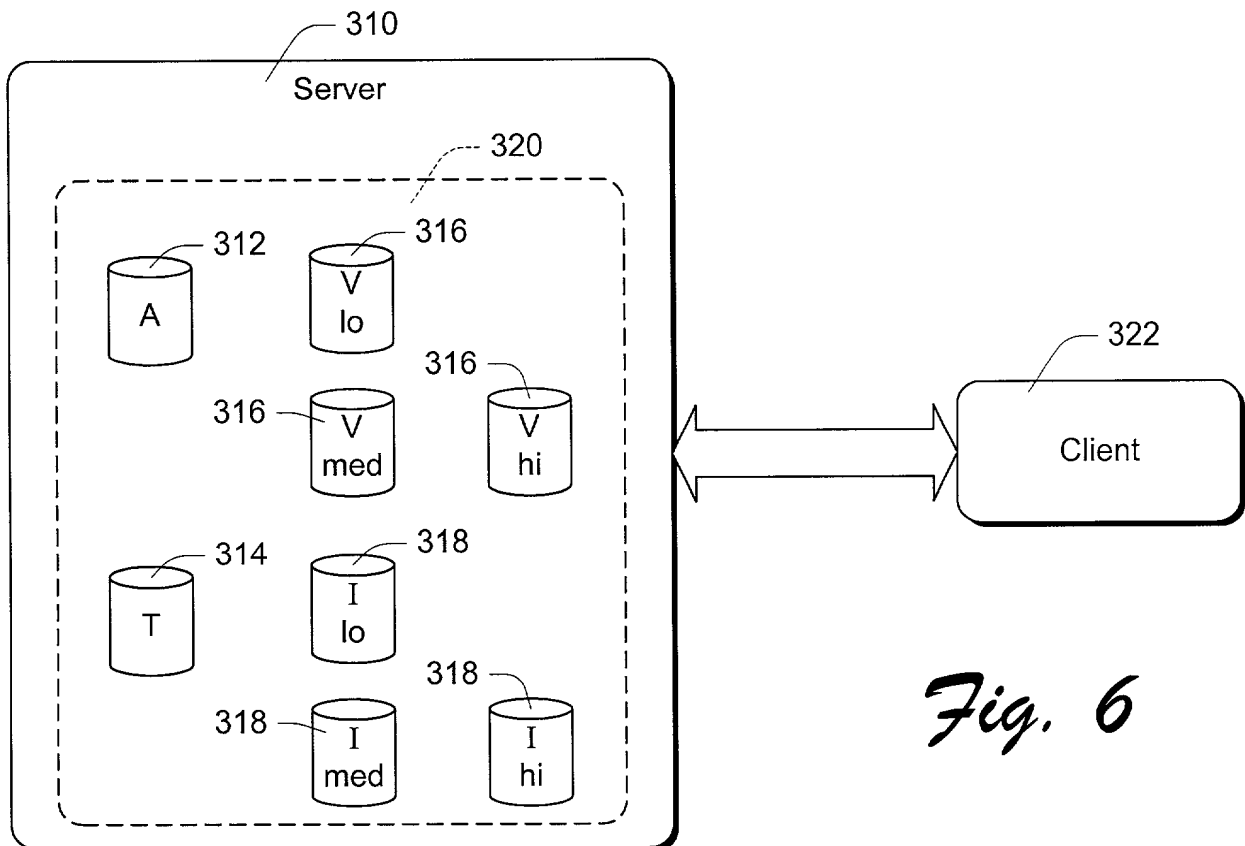
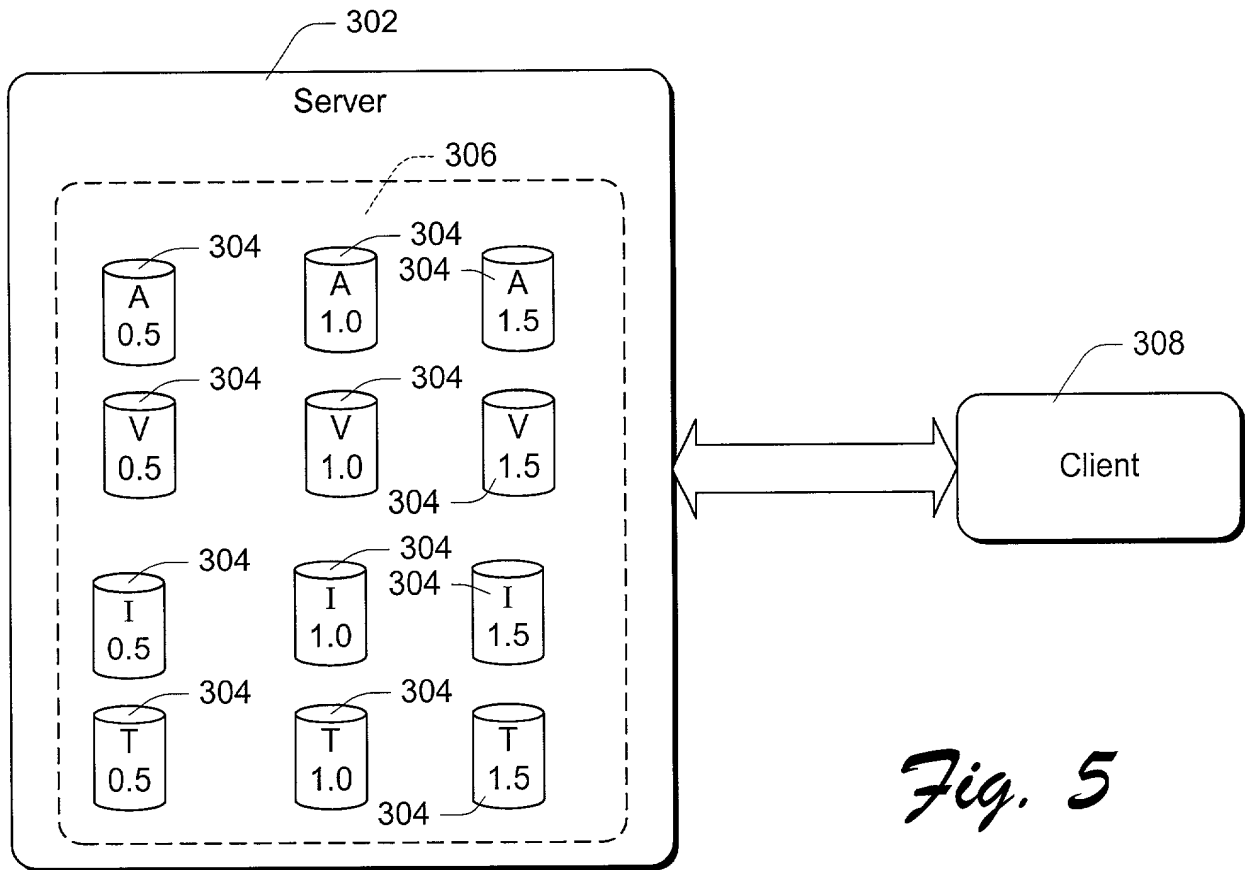
*Fig. 1*

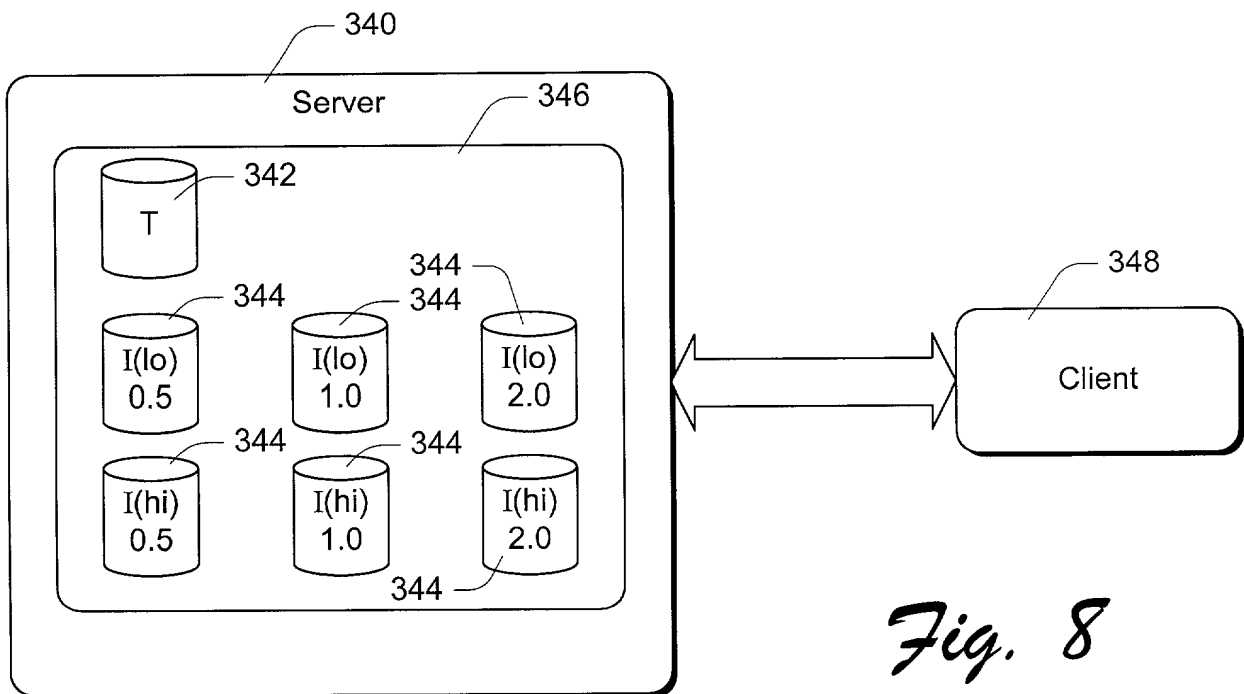
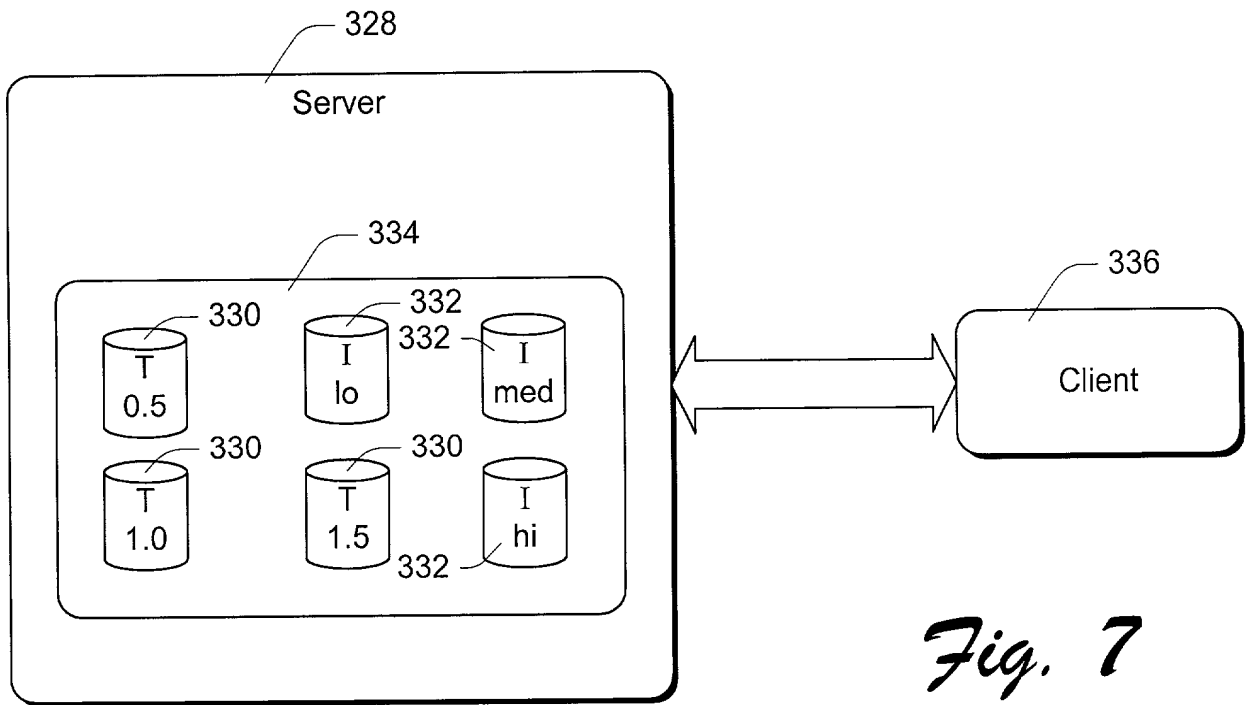


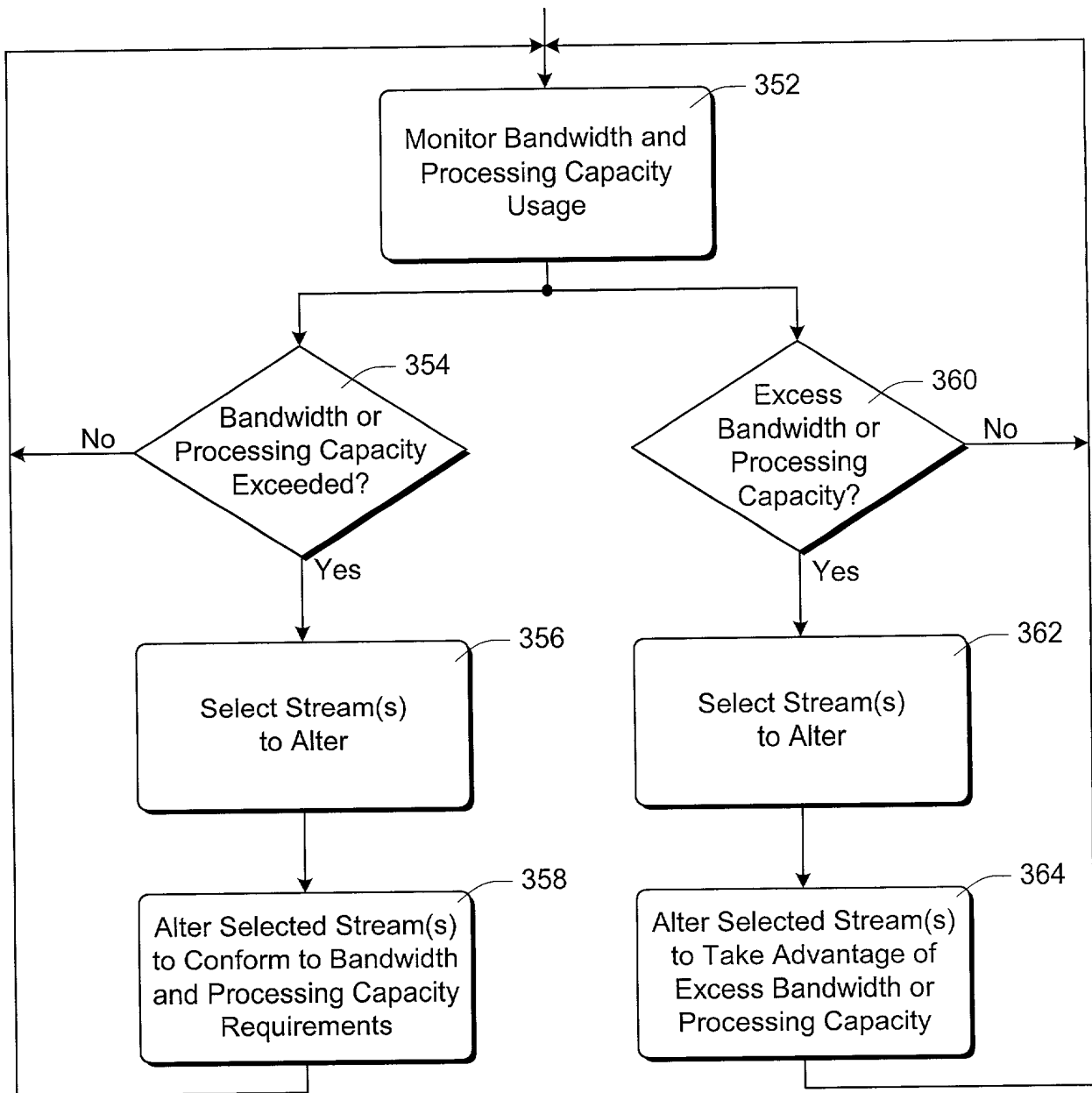
Fig. 3

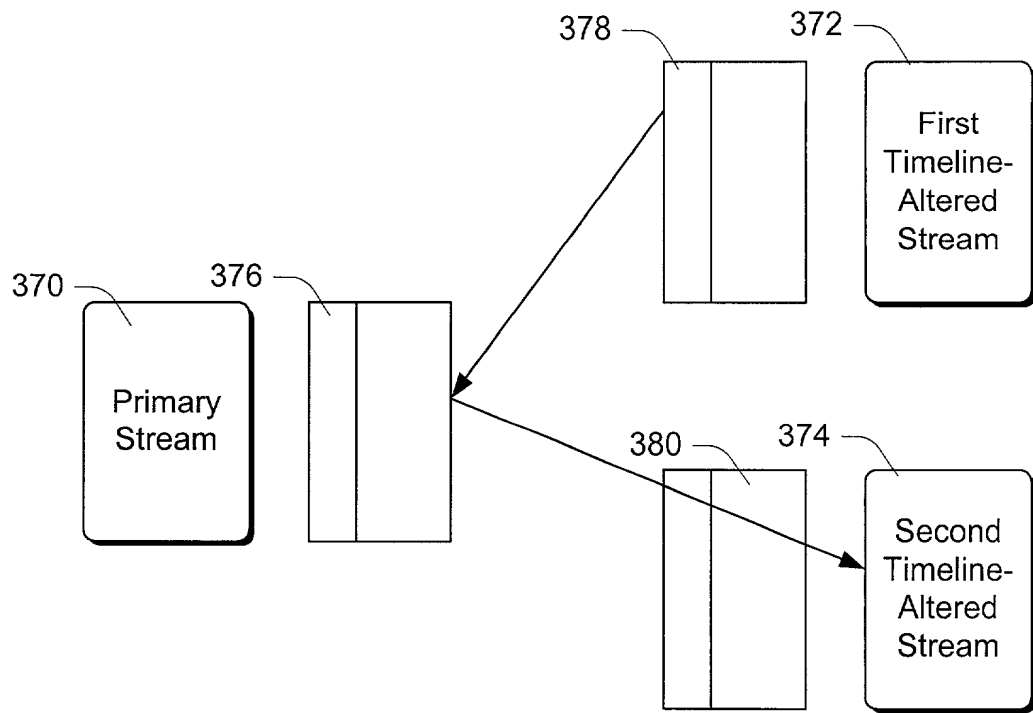


*Fig. 4*

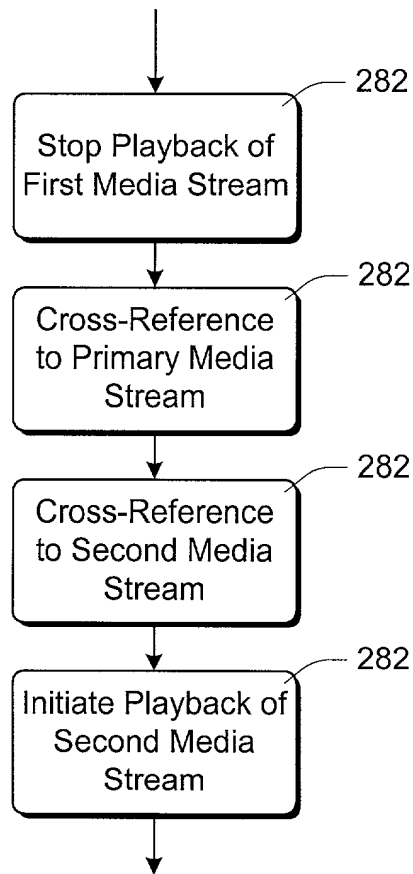




*Fig. 9*



*Fig. 10*

*Fig. 11*

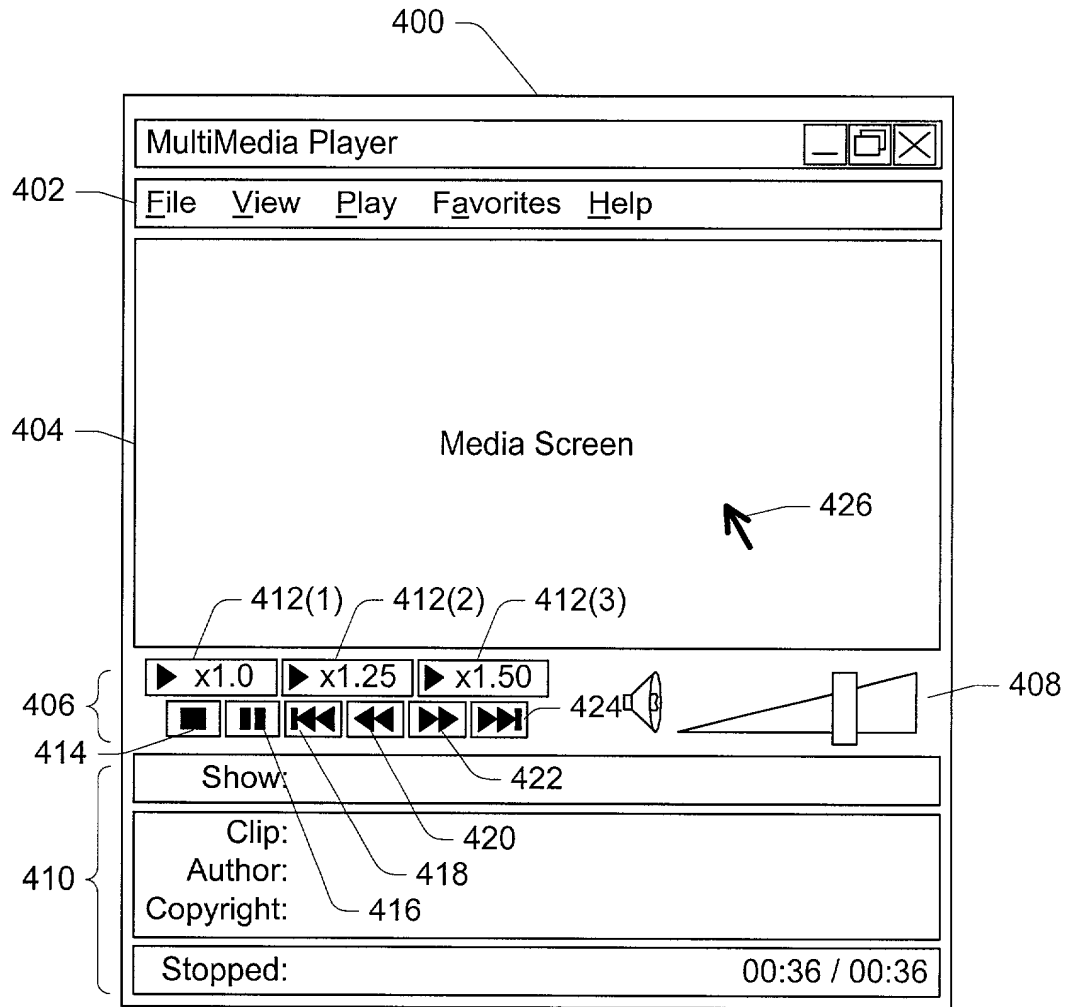
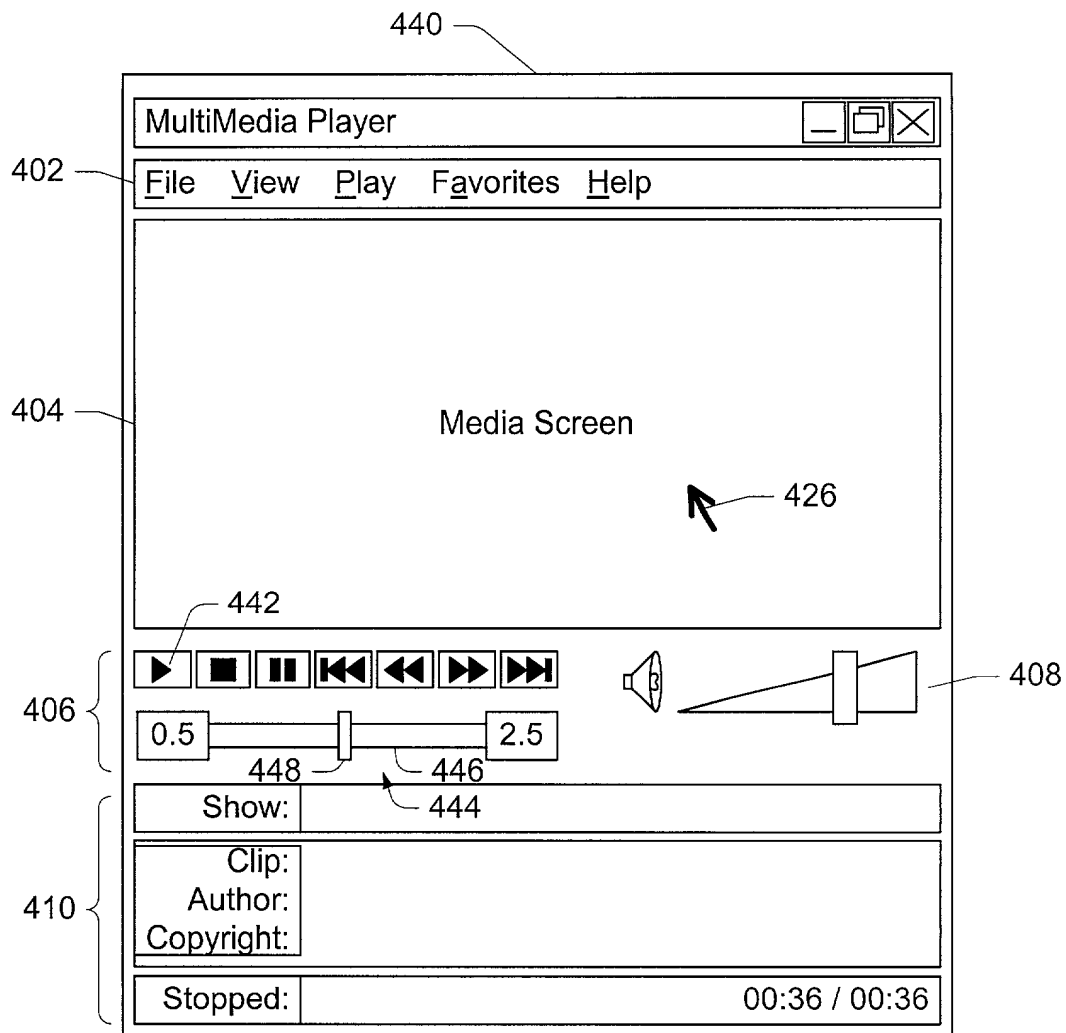


Fig. 12

*Fig. 13*

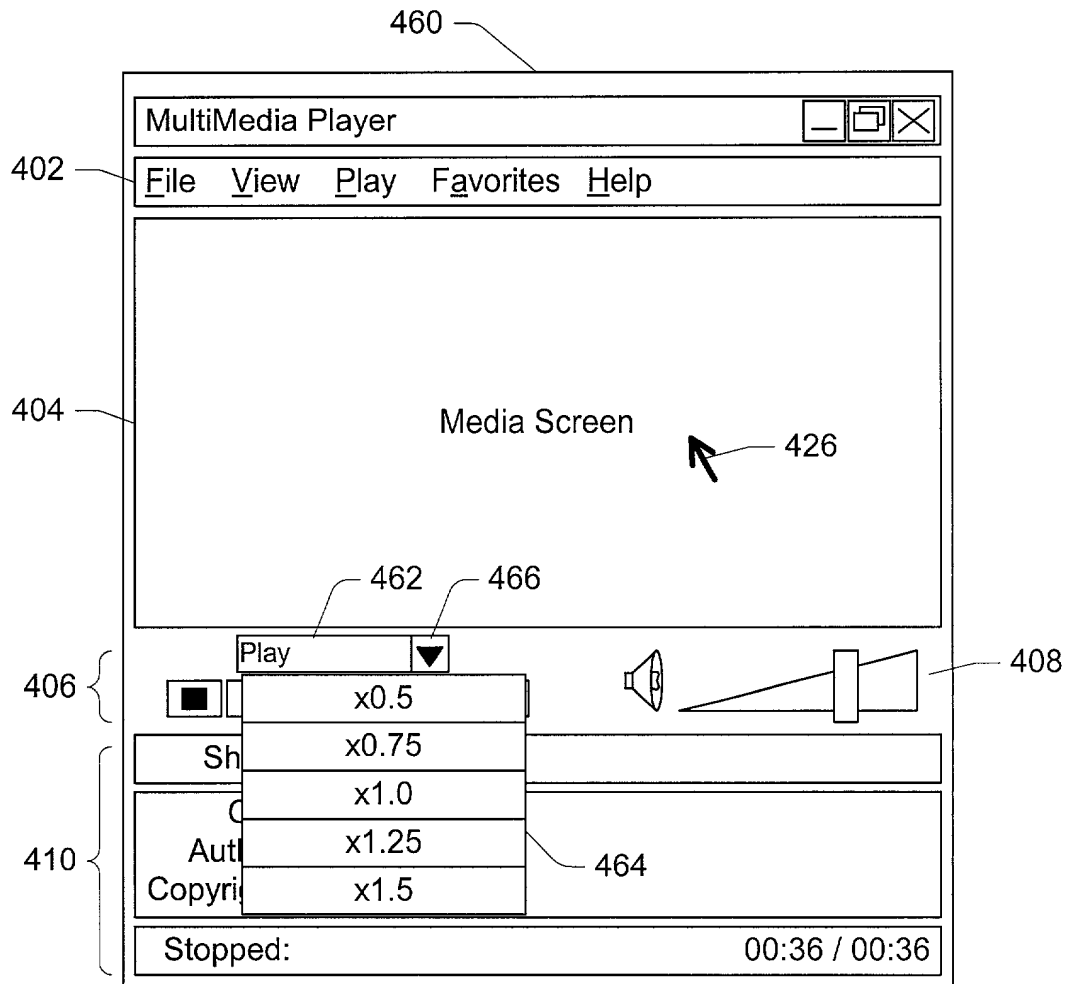


Fig. 14

1 **IN THE UNITED STATES PATENT AND TRADEMARK OFFICE**

2 Inventorship .....Omoigui et al.  
 3 Applicant ..... Microsoft Corporation  
 4 Attorney's Docket No. ....MS1-272USC1  
 Title: Managing Timeline Modification and Synchronization of Multiple Media  
 Streams in Networked Client/Server Systems

5 **DECLARATION FOR PATENT APPLICATION**

6 As a below named inventor, I hereby declare that:

7 My residence, post office address and citizenship are as stated below next to  
 8 my name.

9 I believe I am the original, first and sole inventor (if only one name is listed  
 10 below) or an original, first and joint inventor (if plural names are listed below) of the  
 11 subject matter which is claimed and for which a patent is sought on the invention  
 12 identified above, entitled "Managing Timeline Modification and Synchronization of  
 13 Multiple Media Streams in Networked Client/Server Systems"

14 This application in part discloses and claims subject matter disclosed in my  
 15 earlier filed pending application entitled "Multimedia Timeline Modification in  
 16 Networked Client/Server Systems" which was filed September 15, 1998, having  
 17 application serial number 09/153,664.

18 I have reviewed and understand the content of the above-identified  
 19 specification, including the claims.

20 I claim benefit under Title 35, United States Code, § 120 in connection with  
 21 said earlier filed application.

22 I acknowledge the duty to disclose information which is material to  
 23 patentability as defined in Title 37, Code of Federal Regulations, § 1.56(a), including  
 24 any information which became available between the filing date of the prior  
 25 application and the national or PCT international filing date of this application.

1 PRIOR FOREIGN APPLICATIONS: no applications for foreign patents or  
2 inventor's certificates have been filed prior to the date of execution of this  
3 declaration.

#### 4 **Power of Attorney**

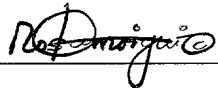
5 I appoint the following attorneys to prosecute this application and transact all  
6 future business in the Patent and Trademark Office connected with this application:  
7 Lewis C. Lee, Reg. No. 34,656; Daniel L. Hayes, Reg. No. 34,618; Allan T.  
8 Sponseller, Reg. 38,318; Steven R. Sponseller, Reg. No. 39,384; James R.  
9 Banowsky, Reg. No. 37,773; Lance R. Sadler, Reg. No. 38,605; Michael A. Proksch,  
10 Reg. No. 43,021; Thomas A. Jolly, Reg. No. 39,241; Kasey C. Christie, Reg. No.  
11 40,559; David A. Morasch, Reg. No. 42,905; Katie E. Sako, Reg. No. 32,628 and  
12 Daniel D. Crouse, Reg. No. 32,022.

13 Send correspondence to: LEE & HAYES, PLLC, 421 W. Riverside Avenue,  
14 Suite 500, Spokane, Washington, 99201. Direct telephone calls to: Allan T.  
15 Sponseller, (509) 324-9256.

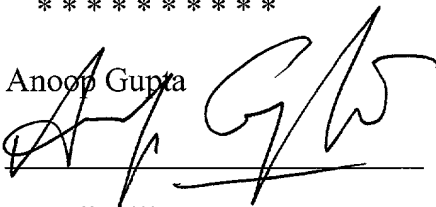
16  
17 All statements made herein of my own knowledge are true and that all  
18 statements made on information and belief are believed to be true; and further that  
19 these statements were made with the knowledge that willful false statements and the  
20 like so made are punishable by fine or imprisonment, or both, under Section 1001 of  
21 Title 18 of the United States Code and that such willful false statement may  
22 jeopardize the validity of the application or any patent issued therefrom.

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25

\*\*\*\*\*

Full name of inventor: Nosakhare D. Omoigui  
Inventor's Signature  Date: 3/24/00  
Residence: Redmond, WA  
Citizenship: Nigeria  
Post Office Address: 8935 160<sup>th</sup> Ave NE #A105  
Redmond, WA 98052

\*\*\*\*\*

Full name of inventor: Anoop Gupta  
Inventor's Signature  Date: 3/28/00  
Residence: Woodinville, WA  
Citizenship: U.S.A.  
Post Office Address: 19908 NE 129<sup>th</sup> Street  
Woodinville, WA 98072